

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a biomedicínského inženýrství**

**Návrh a implementace řídicích algoritmů s PID regulátory na reálném laboratorním modelu s uvažováním nelinearit systému**

**Design and implementation of control algorithms with a PID controller on a real laboratory model with the nonlinearities**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

## Zadání diplomové práce

Student: **Bc. Ondřej Horáček**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2601T004 Měřicí a řídicí technika

Téma: **Návrh a implementace řídicích algoritmů s PID regulátory na reálném laboratorním modelu s uvažováním nelinearit systému**  
**Design and Implementation of Control Algorithms with a PID Controller on a Real Laboratory Model with the Nonlinearities**

### Zásady pro vypracování:

Práce v teoretické části shrnuje nutné podmínky pro úspěšnou implementaci klasického PID regulátoru v průmyslu. Prvním bodem praktické části je kompletně otestovat stávající, firmou Ingeteam a.s. používaný, implementovaný PID regulátor. Hlavním úkolem je na praktickém reálném laboratorním modelu navrhnout, realizovat a otestovat řídicí algoritmy s navrženým PID regulátorem.

### Body zadání:

1. Shrnutí nutných podmínek pro úspěšnou vlastní implementaci klasického PID regulátoru v praxi.
2. Na základě zhodnocení, provedeného v bodě 1, provést kompletní otestování a doladění stávajícího PID regulátoru.
3. Návrh a implementace řídicích algoritmů na reálném laboratorním modelu.
4. Návrh a realizace kaskádní regulace hladiny v nádrži.
5. Návrh a realizace řízení konstantního průtoku kapaliny. Implementace adaptivní linearizační křivky servo ventilu pro řízení konstantního průtoku kapaliny při nelineárně proměnlivém vstupním tlaku kapaliny a nelineární charakteristice vlastního regulačního servo ventilu.
6. Návrh a implementace vizualizace řídicího systému.
7. Otestování kompletního řídicího systému. Vyhodnocení regulačních dějů při různých nastaveních regulovaného systému. Diskuze dosažených výsledků.
8. Zhodnocení a závěr.

### Seznam doporučené odborné literatury:

- [1] BERGER, Hans. *Automating with STEP 7 in STL and SCL*. 5th revised and enlarged edition. Erlangen, Germany:Publicis Publishing, 2009. ISBN 978-3-89578-341-8.
- [2] BERGER, Hans. *Automating with SIMATIC*. 4th edition. Erlangen, Germany: Publicis Publishing, 2009. ISBN 978-3-89578-333-3.
- [3] ASTROM, Karl. J. a Tore HAGGLUND. *PID controllers: Theory, Design, and Tuning*. 2nd Edition. Research Triangle Park(USA) : Instrument Society of America, 1995. 343 s. ISBN 1-55617-516-7.
- [4] O'DWYER, Aidan. *Handbook of PI and PID Controller Tuning Rules*. London(UK):Imperial College Press, c2003. 375 s. ISBN 1-86094-350-0.
- [5] BOBÁL, Vladimír, Josef BÖHM et al. *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace*. 1. vyd. Brno: VUTUIM, 1999, 242 s. ISBN 80-214-1299-2.
- [6] ŠULC, Bohumil a Miluše VÍTEČKOVÁ. *Teorie a praxe návrhu regulačních obvodů*. 1. vyd. Praha: Vydavatelství ČVUT, 2004. 333 s. ISBN 80-01-03007-5.



Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jiří Koziorek, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

doc. Ing. Jiří Koziorek, Ph.D.  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



## Prohlášení

Prohlašuji, že svou diplomovou práci na téma Návrh a implementace řídicích algoritmů s PID regulátory na reálném laboratorním modelu s uvažováním nelinearit systému jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

7.5.2013

V Ostravě dne



Ondřej Horáček

## **Poděkování**

Touto cestou bych chtěl poděkovat vedoucímu mé diplomové práce panu doc. Ing. Jiřímu Koziorkovi, Ph.D. a panu Ing. Jaroslavu Tyšerovi, vedoucímu oddělení automatizace firmy Ingeteam, za poskytnutí cenných rad a konzultací, které mi během tvorby diplomové práce poskytli. Dále bych chtěl také poděkovat zaměstnancům firmy Ingeteam a.s. za pomoc s tvorbou programu mé diplomové práce.

## **Abstrakt**

Práce se zabývá realizací řídicích algoritmů na laboratorním modelu pro průmyslové řízení se systémy PLC. V teoretické části jsou shrnuty podmínky pro implementaci regulátorů, zvláště vícerozměrný regulační obvod v zapojení pro kaskádní regulaci. Dále je v práci popis laboratorního zařízení „Vodní svět,“ na kterém byla celá práce realizována. Hlavním tématem je potlačení nelinearity proporcionálního ventilu. K tomu byla vytvořena linearizační funkce, která za pomoci lineární interpolace vstupních dat vypočítává hodnotu otevření ventilu, pro kterou odpovídá požadovaný průtok v závislosti na tlaku, který je na vstupní straně proporcionálního ventilu. Dalším řešením potlačení nelinearity, byla realizace kaskádní regulace, kdy do obvodu byla přidána další regulovaná veličina – průtok. Srovnání kaskádní regulace, regulace s pomocí linearizační funkce a přímé regulace je provedeno na konci práce.

## **Klíčová slova**

PID regulátor, PLC, kaskádní regulace, linearizace, proporcionální ventil, SIMATIC S7

## **Abstract**

The work deals with the implementation of control algorithms on a laboratory model for industrial control systems PLC. In the theoretical section are summarized the conditions for the implementation of the controller, especially multidimensional control circuit connected for cascade control. It is the job description of laboratory equipment "Water World," in which all the work was realized. The main issue is the suppression of nonlinearity proportional valve. It was created linearization block using linear interpolation of input data for compute the valve opening, which corresponds to the set flow rate depending on the pressure, which is on the input of the proportional valve. Another solution for suppression of nonlinearity is the implementation of cascade control, when was added to the circuit another controlled variable – flow. Comparison of cascade control, linearizing control block and direct control is performed at the end of the work.

## **Key words**

PID control, PLC, cascade control, linearization, proportional valve, SIMATIC S7

## Seznam symbolů a zkratk

CPU .....	Centrální procesorová jednotka, procesor
D složka .....	Derivační složka regulátoru
DB .....	Data block, blok s daty
DOBD .....	Dopředná obdélníková metoda
FBD .....	Function block diagram, grafický programovací jazyk
Floating point .....	Datový formát, čísla s desetinnou čárkou
I složka .....	Integrační složka regulátoru
INT .....	Datový formát, celá čísla
IP adresa .....	Internet protokol, adresa zařízení v síti
LICHO .....	Lichoběžníková obdélníková metoda
P složka .....	Proporcionální složka regulátoru
PLC .....	Programable logic controller, programovatelný automat
PSD .....	Proportional summation differentiation, diskretní verze PID regulátoru
ZOBD .....	Zpětná obdélníková metoda



## Seznam obrázků

Obr. 1 Jednoduchý regulační obvod.....	2
Obr. 2 Obdélníkové metody .....	5
Obr. 3 Vícerozměrný regulační obvod - kaskádní regulace .....	7
Obr. 4 Nádrž s regulovanou výškou hladiny .....	9
Obr. 5 Schéma nádrže .....	10
Obr. 6 Model nádrže s nuceným odtokem .....	11
Obr. 7 Model nádrže v Simulinku .....	12
Obr. 8 Odezva na vstupní průtok $2,4\text{m}^3/\text{h}$ .....	12
Obr. 9 Nástroj Linear Analysis .....	13
Obr. 10 Zjištění pracovního bodu pro linearizaci.....	13
Obr. 11 Linearizovaný model.....	14
Obr. 12 Porovnání původního a linearizovaného systému .....	14
Obr. 13 Nástroj PID Tuner .....	15
Obr. 14 Zpětnovazební regulační obvod pro regulaci výšky hladiny .....	15
Obr. 15 Regulační děj .....	16
Obr. 16 PID regulátor.....	17
Obr. 17 „Vodní svět“.....	19
Obr. 18 Snímač tlaku D2411 .....	20
Obr. 19 Snímač tlaku LMK351 .....	21
Obr. 20 Indukční průtokoměr.....	21
Obr. 21 Proud pro $0 \gg +Q$ .....	22
Obr. 22 Proud pro $0 \gg -Q$ .....	22
Obr. 23 Proud pro $ Q $ .....	22
Obr. 24 Proud pro $-Q \gg +Q[7]$ .....	22
Obr. 25 Pohon SQX62 .....	23
Obr. 26 Polynomická interpolace.....	25
Obr. 27 Lagrangeův interpolační polynom .....	26
Obr. 28 Lineární interpolace .....	28
Obr. 29 Průtočná charakteristika proporcionálního ventilu .....	29
Obr. 30 Ventil a tlakoměr pro měření tlaku před ventilem .....	30
Obr. 31 Charakteristika tlaku na vstupu ventilu.....	31
Obr. 32 Ruční ventil pro změnu tlaku před proporcionálním ventilem .....	32
Obr. 33 Vykreslení naměřených dat.....	34
Obr. 34 Data aproximovaná polynomem 2. řádu .....	36
Obr. 35 Závislost otevření na průtoku pro jednotlivé tlaky.....	38
Obr. 36 Struktura datového bloku, z kterého funkce hledá hodnoty.....	39
Obr. 37 Algoritmus výpočtu výstupní hodnoty funkce .....	41
Obr. 38 Vzhled vytvořené linearizační funkce v FBD.....	48
Obr. 39 Grafické znázornění výpočtu výstupu funkce.....	49
Obr. 40 Hardwarová konfigurace .....	50
Obr. 41 Vizualizace systému.....	56
Obr. 42 Analogové měření veličin v potrubí.....	56
Obr. 43 Analogové měření stavu obou nádrží.....	57
Obr. 44 Okno pro analogové měření .....	57

Obr. 45 Bloky pro ruční ovládání zařízení .....	58
Obr. 46 Bloky pro ovládání čerpadla a proporcionálního ventilu .....	59
Obr. 47 Ovládání regulace .....	60
Obr. 48 Blok regulátoru .....	60
Obr. 49 Blokové schéma regulace pomocí otevření ventilu .....	61
Obr. 50 Odezva při regulaci pomocí otevření ventilu .....	62
Obr. 51 Odezva na změnu o 10mm při různých průtocích při regulaci otevření ventilu .....	63
Obr. 52 Blokové schéma kaskádní regulace .....	64
Obr. 53 Přechodová charakteristika kaskádní regulace.....	64
Obr. 54 Odezva na změnu o 10mm při různých průtocích při kaskádní regulaci .....	65
Obr. 55 Blokové schéma regulace s použitím linearizační funkce .....	66
Obr. 56 Přechodová charakteristika regulace s použitím linearizační funkce.....	67
Obr. 57 Odezva na změnu o 10mm při různých průtocích při použití linearizační funkce.....	68

## Seznam tabulek

Tab. 1 Výpočet parametrů pomocí zvolené obdélníkové metody .....	6
Tab. 2 Vstupy PID regulátoru .....	18
Tab. 3 Možnosti nastavení z vizualizace .....	18
Tab. 4 Příkazy z vizualizace .....	18
Tab. 5 Technické parametry školícího zařízení .....	20
Tab. 6 Parametry pohonu SQX62 .....	24
Tab. 7 Tabulka naměřených hodnot průtoku v závislosti na otevření ventilu .....	29
Tab. 8 Tabulka naměřených hodnot vstupního tlaku v závislosti na změně otevření .....	31
Tab. 9 Naměřená data .....	33
Tab. 10 Data aproximovaná polynomem 2.řádu .....	35
Tab. 11 Závislost otevření na průtoku při jmenovitém tlaku .....	37
Tab. 12 Závislost otevření ventilu na průtoku a vstupním tlaku .....	38
Tab. 13 Struktura dat uložených v DB .....	40
Tab. 14 Nalezené řádky pro uvedený příklad .....	44
Tab. 15 Nalezené sloupce pro uvedený příklad .....	44
Tab. 16 Nalezené hodnoty otevření pro uvedený příklad .....	45
Tab. 17 Význam proměnných použitých ve výpočtu .....	45
Tab. 18 Postup výpočtu výstupní hodnoty $Y$ .....	46
Tab. 19 Schématické znázornění výstupů linearizační funkce .....	47
Tab. 20 Tabulka vstupů a výstupů funkce .....	48
Tab. 21 Adresy jednotky A3 .....	51
Tab. 22 Adresy jednotky A4 .....	51
Tab. 23 Adresy jednotky A5 .....	51
Tab. 24 Adresy jednotky A7 .....	52
Tab. 25 Adresy jednotky A8 .....	52
Tab. 26 Adresy jednotky A9 .....	52
Tab. 27 Adresy jednotky A10 .....	52
Tab. 28 Adresy jednotky A11 .....	52
Tab. 29 Adresy jednotky A12 .....	53
Tab. 30 Struktura příkazů .....	55
Tab. 31 Struktura stavů .....	55
Tab. 32 Měřené veličiny .....	58
Tab. 33 Signály regulace pomocí otevření ventilu .....	61
Tab. 34 Signály kaskádní regulace .....	64
Tab. 35 Signály regulace s použitím linearizační funkce .....	66



## Obsah

Úvod.....	1
1. Spojité regulátory .....	2
1.1. Jednoduchý regulační obvod .....	2
1.2. Jednoduché spojitý regulátory .....	3
1.3. Složené regulátory .....	4
2. Číslicové regulátory .....	5
2.1. Diskretizace PID regulátoru .....	5
3. Kaskádní regulace .....	7
4. Linearizace vodní nádrže s odtokem .....	9
4.1. Identifikace regulované soustavy .....	9
4.2. Linearizace pomocí nástroje Linear Analysis .....	12
5. PID vyvinutý firmou INGETEAM .....	17
6. Popis školícího zařízení „Vodní svět“ .....	19
6.1. Obecný popis .....	19
6.2. Snímače relativního tlaku s keramickou membránou D2411 .....	20
6.3. Snímače tlaku s keramickou čelní membránou LMK 351 .....	20
6.4. Indukční průtokoměr FLONET FN 20XX.1 .....	21
6.5. Ventil RV103 s elektrickým pohonem SQX62 .....	23
7. Linearizační funkce pro potlačení nelinearity proporcionálního ventilu .....	24
7.1. Interpolace funkce .....	24
7.2. Naměřená data .....	28
7.3. Vstupní data pro linearizační funkci .....	32
7.4. Návrh linearizační funkce v prostředí STEP7 .....	39
8. Hardwarová konfigurace .....	50
8.1. Decentrální periferie SIMATIC ET 200S .....	50
9. Řídicí program v PLC .....	54
9.1. UserProgram .....	54
10. Vizualizace .....	56
10.1. Regulace průtokem .....	59
10.2. Regulace otevření ventilu .....	59
11. Zhodnocení výsledků .....	61
11.1. Regulace pomocí otevření ventilu .....	61
11.2. Kaskádní regulace .....	63

11.3.	Použití linearizační funkce .....	66
Závěr .....		69

# Úvod

Regulace strojů a zařízení se považuje za nejstarší a nejmasovější uplatnění teoreticky a matematicky popsaných principů, které jsou součástí vědního oboru kybernetika.

Tato práce byla realizována ve firmě Ingeteam a.s. na školícím zařízení „Vodní svět.“ Jedná se o zařízení pro simulaci průmyslových procesů. Podrobný popis zařízení je uveden v práci. Na zařízení byla realizována regulace výšky hladiny pomocí změny otevření proporcionálního ventilu. Jeho charakteristika je však velmi nelineární a od otevření 50% se průtok zařízením téměř nemění.

Proto byly navrženy dvě řešení, jak potlačit nelinearitu ventilu. Prvním řešením je kaskádní regulace. Do obvodu byl přidán další sekundární regulátor, který reguluje průtok potrubními trasami zařízení. Primární regulátor už nereguluje výšku hladiny pomocí otevření ventilu, ale pomocí průtoku, respektive přítoku množství vody do nádrže. Sekundární regulátor by měl v momentu, kdy bude změněna rychlost odtoku z nádrže, zajistit stejně rychlou reakci na změnu požadované výšky hladiny v celém rozsahu požadovaného průtoku. V přímé regulaci, kdy bude výška hladiny regulována otevřením ventilu, je při zvýšení průtoku reakce pomalejší. To je dáno tím, že změna o např. 5% při otevření 20% neznamena stejnou změnu průtoku jako při otevření 30%. Proto je reakce jiná v různých pracovních bodech. Od sekundárního regulátoru je očekáváno, že tuto nelinearitu odstraní. Výsledky kvality regulace jsou uvedeny v závěru práce.

Druhým řešením byla linearizace ventilu. Proto byl vytvořen blok, který na základě naměřených dat a požadovaného průtoku vypočte otevření ventilu, které požadovanému průtoku odpovídá. Požadavkem bylo, aby byl blok schopen reagovat na změnu tlaku na vstupu ventilu. Proto byl přidán další vstup, na který je přiveden signál z tlakoměru, který je umístěn před ventilem. Pro výpočet otevření byla použita lineární interpolace.

Pro správnou funkci ventilu, byla naměřena data, kdy byl při různých velikostech otevření ručním ventilem snižován tlak a měřen průtok, který zařízením protéká. Úprava těchto dat pro vstupní funkci je v práci popsána také.

Závěr práce se zabývá shrnutím uvedených řešení.

Jako řídicí systém byl použit programovatelný automat SIMATIC S7 400 od firmy Siemens. Program byl vytvořen v grafickém jazyce FBD. Linearizační funkce byla vytvořena v jazyce STL.



## 1. Spojité regulátory

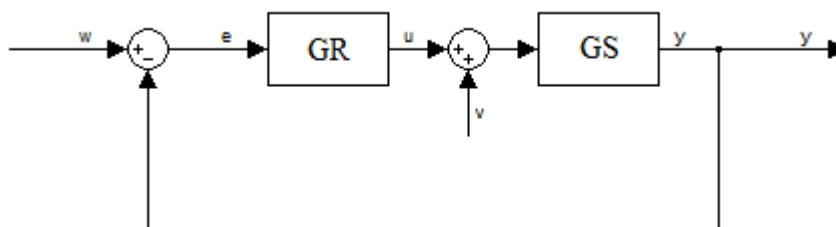
Bezkonkurenčně nejpoužívanější regulátory používané v průmyslu jsou regulátory typu PID. Jedná se o spojité regulátory s proporcionální, integrační a derivační složkou.

Historie PID regulátorů sahá až do devadesátých let devatenáctého století. Jeden z prvních příkladů užití regulátorů byl vyvinut Američanem Elmerem Sperryem v roce 1911. První publikovaná teoretická analýza PID regulátoru byla vydána v roce 1922. Autorem byl rus Nicolas Minorsky, který navrhoval automatické řízení systémů pro americké námořnictvo. Na základě pozorování kormidelníka uvádí, že kormidelník ovládá loď nejen na základě aktuální chyby, ale také na základě chyby minulé a na rychlosti změny. Poté byla provedena matematická analýza podle Minorského, jejíž cílem byla stabilita, ne obecná kontrola. Zatímco proporcionální řízení zajišťovalo stabilitu proti malým poruchám, ale bylo nedostatečné proti stabilním poruchám, například vichřici. To vyžadovalo přidání integrační složky. Nakonec byla přidána derivace pro zlepšení stability.

Zkoušky byly provedeny na lodi USS New Mexico s regulátorem ovládající úhlovou rychlost kormidla. Výsledky byly lepší, než jakých mohla většina kormidelníků dosáhnout.

Většina regulátorů používaných dnes v průmyslu obsahuje proporcionální a integrační složku (PI regulátor). Všechny stabilní procesy můžeme řídit I regulátorem. Přidáním P složky chování regulátoru ještě vylepšíme. PI regulátor se hodí pro všechny soustavy 1. řádu. Derivační složka se užívá méně často, mnohdy je vypnutá a zapínáme ji tehdy, pokud se jedná o soustavy vyššího řádu, soustavy se zpožděním, nebo tam, kde jsou kladeny větší nároky na řízení (přesné, optimální, robustní).[2]

### 1.1. Jednoduchý regulační obvod



Obr. 1 Jednoduchý regulační obvod

Regulátor společně s regulovanou soustavou je zapojen do uzavřené smyčky se zápornou zpětnou vazbou. Od žádané hodnoty  $w$  je odečtena skutečná hodnota  $y$ . Výsledkem je regulační odchylka, které vstupuje do samotného výpočetního procesu regulátoru.

$$u(t) = \underbrace{r_0 \cdot e(t)}_P + \underbrace{r_i \cdot \int_{t_0}^t e(t) dt}_I + \underbrace{r_d \cdot \frac{de(t)}{dt}}_D \quad (1)$$

Výstupem regulátoru je akční zásah  $u$ , který přímo zasahuje do regulované soustavy. Regulovaná veličina  $y$  v každém okamžiku musí co nejpřesněji sledovat požadovanou hodnotu  $w$  a to nezávisle na působení poruchových veličin  $v$  a  $n$  a na změnách dynamických vlastností řízeného systému. To můžeme rozdělit na požadavky na kvalitu sledování požadované hodnoty, potlačení vlivu

poruchových hodnot a na požadavek robustnosti vzhledem ke stabilitě a kvalitě řízení. Splnění těchto požadavků zajišťuje regulátor, který generuje akční veličinu  $u$  na základě regulační odchylky  $e$ .

Při řízení reálných soustav narážíme na řadu principiálních a technických problémů. Prvním faktem, který souvisí s návrhem jakéhokoli typu regulátoru je, že každá reálná soustava je nelineární. Záleží na konkrétní situaci a typech nelinearit, které se u dané soustavy vyskytují. Proto existují metody, pomocí kterých lze za definovaných podmínek dosáhnout toho, že na regulační obvod můžeme poté aplikovat metody, které jsou určeny pro lineární obvody. Druhým faktem je, že realizace regulátoru na konkrétní platformě (hardwaru) zřídka vede na práci se spojitými tvary pro popis činnosti regulátoru. Třetím důležitým faktem při návrhu regulátoru je, že zdaleka ne u všech systémů se podaří odvodit matematický model odpovídající vnějšímu nebo vnitřnímu popisu systému. V takovém případě je nutno přistoupit k tzv. identifikaci, což vede na nalezení aproximovaného matematického modelu, případně jazykového popisu reálného systému za použití zvolené vhodné metody.[1]

## 1.2. Jednoduché spojité regulátory

### P regulátor

Nejjednodušším regulátorem je proporcionální regulátor. Akční veličina na výstupu je přímo úměrná regulační odchylce na vstupu.

Akční veličinu lze tedy vyjádřit vztahem:

$$u(t) = r_0 \cdot e(t) \quad (2)$$

Výhodou proporcionálního regulátoru je jeho jednoduchost. Nevýhoda je, že pracuje s trvalou regulační odchylkou. Ta je způsobena jeho matematickými vlastnostmi (nulová regulační odchylka = nulový akční zásah). Trvalou regulační odchylku tedy nelze odstranit, můžeme pouze ovlivnit její velikost na úkor stability. Chceme-li, aby regulátor pracoval bez trvalé regulační odchylky, musíme zvolit jiný typ regulátoru.[1]

### I regulátor

Každé hodnotě vstupní veličiny odpovídá úměrná změna rychlosti výstupní veličiny. Integrační regulátor má při nulovém vstupním signálu (regulační odchylce) nenulový výstup. To znamená, že jeho nejvýznamnější vlastností je, že pracuje bez trvalé regulační odchylky.[1]

Akční veličinu na výstupu lze vyjádřit vztahem:

$$u(t) = r_i \cdot \int_{t_0}^t e(t) dt \quad (3)$$

### D regulátor

Derivační regulátor nelze samostatně fyzikálně realizovat. Proto se používá pouze ve složených regulátorech PD nebo PID. Výstup regulátoru (akční veličina) je přímo úměrný derivaci regulační odchylky a nese informaci o jejích budoucích hodnotách. Zavádí se pro urychlení přechodového děje a do určité míry předvídá změny, které teprve nastanou.[1]

Akční veličinu na výstupu lze vyjádřit vztahem:

$$u(t) = r_d \cdot \frac{de(t)}{dt} \quad (4)$$

### 1.3. Složené regulátory

Pro zlepšení regulačního děje, se zavádí složené regulátory. Jedná se o kombinace jednoduchých regulátorů. Výsledkem je to, že odstraňují jejich nevýhody a přispívají ke stabilitě a kvalitě regulačního děje.

#### **PI regulátor**

U PI regulátoru jsou v akční veličině zastoupeny dvě složky, proporcionální a integrační. Proporcionální je úměrná velikosti regulační odchylky a integrační jejímu integrálu. Do regulačního děje zasáhne nejprve proporcionální složka a teprve potom složka integrační. Oproti P regulátoru tedy pracuje bez trvalé regulační odchylky.[1]

Jedná se o nejpoužívanější regulátor vůbec.

Akční veličinu na výstupu lze vyjádřit vztahem:

$$u(t) = r_0 \cdot e(t) + r_i \cdot \int_{t_0}^t e(t) dt \quad (5)$$

#### **PD regulátor**

Obdobně jako PI regulátor je PD regulátor složen ze dvou složek, proporcionální a derivační. Jedna je úměrná regulační odchylce, druhá její derivaci. Do děje nejprve zasáhne derivační složka, která celý děj nejprve urychlí, a poté zasáhne proporcionální složka, která děj stabilizuje. Protože není přítomná integrační složka, regulátor pracuje s trvalou regulační odchylkou. [1]

Akční veličinu na výstupu lze vyjádřit vztahem:

$$u(t) = r_0 \cdot e(t) + r_d \cdot \frac{de(t)}{dt} \quad (6)$$

#### **PID regulátor**

PID regulátor má ve výstupním signálu obsaženy všechny tři složky, které jsou úměrné velikosti regulační odchylky, jejímu integrálu a její derivaci. Do regulačního děje nejprve zasáhne derivační složka, později proporcionální složka a teprve na závěr integrační složka. PID regulátor pracuje bez trvalé regulační odchylky a lze jej hodnotit jako nejdokonalejší spojitý regulátor.[1]

Akční veličinu na výstupu lze vyjádřit vztahem:

$$u(t) = r_0 \cdot e(t) + r_i \cdot \int_{t_0}^t e(t) dt + r_d \cdot \frac{de(t)}{dt} \quad (7)$$



## 2. Číslicové regulátory

Jak už bylo napsáno výše, faktem je, že realizace regulátoru na konkrétní platformě (hardwaru) zřídka vede na práci se spojitými tvary pro popis činnosti regulátoru. Proto se používají číslicové regulátory.

Od číslicového regulátoru očekáváme stejnou funkci jako od spojitého regulátoru a to vstupující regulační odchylku zesilovat, integrovat a derivovat. Proto se při sestavování algoritmu pro číslicový regulátor vychází z funkce a tím i rovnice spojitého PID regulátoru. Jeho diskretní verze se označuje PSD regulátor. V současné době se PSD nachází ve většině řídicích systémů.[3]

### 2.1. Diskretizace PID regulátoru

Máme spojitý regulátor s konstantami P, I a D. přenos regulátoru ve standardní formě je:

$$G(s) = K_p \cdot \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (8)$$

Parametry regulátoru jsou:

$$K_p = P \quad (9)$$

$$T_i = \frac{P}{I} \quad (10)$$

$$T_d = \frac{D}{P} \quad (11)$$

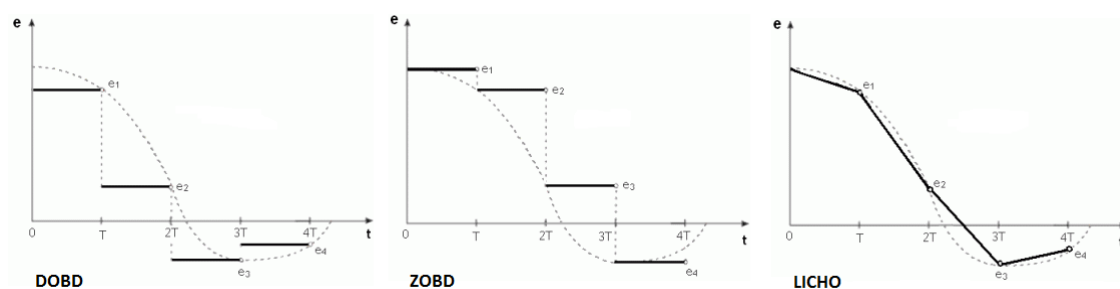
Přenos PSD regulátoru je dán tvary:

$$G_{PSD}(z) = \frac{d_0 + d_1 z^{-1} + d_2 z^{-2}}{1 - z^{-1}} \quad (12)$$

$$G_{PS}(z) = \frac{d_0 + d_1 z^{-1}}{1 - z^{-1}} \quad (13)$$

$$G_{PD}(z) = d_0 + d_1 z^{-1} \quad (14)$$

Nyní zvolíme obdélníkovou metodu, pomocí které vypočítáme konstanty.



Obr. 2 Obdélníkové metody

DOBD – dopředná obdélníková metoda

ZOBD – zpětná obdélníková metoda

LICH – lichoběžníková obdélníková metoda

Na základě zvolené metody můžeme stanovit koeficienty PSD regulátoru pomocí tabulky.

	<b>DOBD</b>	<b>ZOBD</b>	<b>LICHO</b>
$d_0$	$K_p \cdot \left(1 + \frac{T_d}{T_0}\right)$	$K_p \cdot \left(1 + \frac{T_0}{T_i} + \frac{T_d}{T_0}\right)$	$K_p \cdot \left(1 + \frac{T_0}{2 \cdot T_i} + \frac{T_d}{T_0}\right)$
$d_1$	$-K_p \cdot \left(1 - \frac{T_0}{T_i} + \frac{2 \cdot T_d}{T_0}\right)$	$-K_p \cdot \left(1 + \frac{2 \cdot T_d}{T_0}\right)$	$-K_p \cdot \left(1 - \frac{T_0}{2 \cdot T_i} + \frac{2 \cdot T_d}{T_0}\right)$
$d_2$	$K_p \cdot \frac{T_d}{T_0}$	$K_p \cdot \frac{T_d}{T_0}$	$K_p \cdot \frac{T_d}{T_0}$

Tab. 1 Výpočet parametrů pomocí zvolené obdélníkové metody

$T_0$  – perioda vzorkování

Akční veličinu lze vyjádřit vztahem:

$$u(k) = d_0 e(k) + d_1 e(k-1) + d_2 e(k-2) + u(k-1) \quad (15)$$

Kde:

$e(k)$  – regulační odchylka

$e(k-1)$  – minulá regulační odchylka

$e(k-2)$  – předminulá regulační odchylka

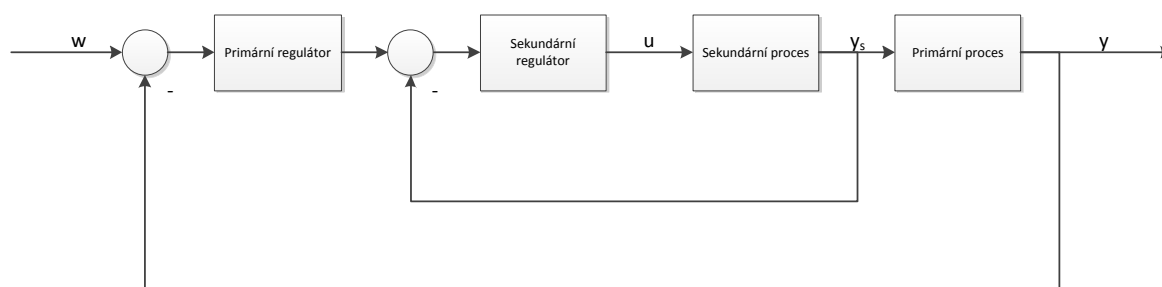
$u(k-1)$  – minulý akční zásah

### 3. Kaskádní regulace

Existuje několik vcelku snadných postupů umožňující zlepšení kvality regulace

- Zavedení D složky – vede k méně kmitavým odezvám
- Zavedení dopředné vazby - včasná reakce na vnější poruchy, dovedeme-li je měřit
- Kompenzace dopravního zpoždění – pro systémy s významným dopravním zpožděním
- Omezení vlivu nelinearit – Lineární obvody se lépe regulují
- Kaskádní regulace

Pokud můžeme regulovaný obvod rozdělit na dvě samostatné části, je kaskádní regulace užitečná a přínosná. První část bývá s rychlejší dynamikou a druhá část s dynamikou pomalejší nebo s dopravním zpožděním.



Obr. 3 Vícerozměrný regulační obvod - kaskádní regulace

Základním předpokladem pro kaskádní regulaci je, že můžeme měřit výstupy obou částí regulovaného procesu  $y$  a  $y_s$ . Pro takový proces lze použít klasický jednosmyčkový regulační obvod s regulátorem pro  $y$  jako regulovanou veličinu  $u$  jako akční zásah. Kaskádní uspořádání však způsobí podstatně rychlejší a kvalitnější potlačení vlivů poruch působící na jednu nebo obě části procesu.

V kaskádní regulaci využijeme obě měřené veličiny jednoho procesu, což představuje větší informaci než při klasické regulaci. Použití větší informace se projeví ve zlepšení kvality regulace. Častokrát to je více než desetinásobně menší přeregulování při kompenzaci poruchy a více než 3krát kratší doba ustálení oproti klasické jednosmyčkové regulaci

Základní výhody dvousmyčkové kaskádní regulace oproti klasické jednosmyčkové regulaci jsou tři.

První výhodou je, že porucha, která působí na sekundární proces, je potlačena sekundárním regulátorem před tím, než začne ovlivňovat primární proces. Účinek na primární regulovanou veličinu tak bude minimální.

Druhou výhodou je, že se urychlí reakce sekundárního procesu.

Třetí výhodou je, že sekundární regulátor tlumí vliv změny zesílení sekundárního procesu a tak zlepšuje celkovou robustnost regulace.

V případě dvousmyčkové kaskádní regulace je zejména vhodné, když v sekundárním obvodu působí hlavní poruchy celého regulačního obvodu, neboť je možné poměrně rychle a efektivně potlačit jejich vliv.

Sekundární obvod také může obsahovat členy s proměnným zesílením. Je rovněž vhodné, je-li sekundární smyčka rychlejší než primární. Poměr doby ustálení regulačních odezev v primární smyčce proti době ustálení v sekundární smyčce má být přinejmenším 5.

Integrační část PID regulátoru obvykle obsahuje pouze primární regulátor. Proporcionální zesílení sekundárního regulátoru totiž bývá větší vzhledem k možnosti potlačit vliv poruch působících na sekundární proces. Ty se potom k primárnímu procesu nedostanou. Navíc trvalá regulační odchylka v sekundárním obvodu není příliš důležitá, neboť regulace výstupu sekundárního obvodu není cílem kaskádní regulace. Sekundárním regulátorem je proto obvykle pouze P regulátor s větším zesílením.

Primárním regulátorem je obvykle PI nebo PID regulátor. Integrační část primárního regulátoru bude potlačovat vliv poruch působících nejen v sekundárním obvodu, ale také v primárním obvodu. Derivační část v souladu s jejím příznivým působením odezvu systému urychlí a stabilizuje.[4]

#### 4. Linearizace vodní nádrže s odtokem

##### 4.1. Identifikace regulované soustavy

Regulovanou soustavou v úloze je nádrž na školicím zařízení „Vodní svět“.



Obr. 4 Nádrž s regulovanou výškou hladiny

Nejprve je soustavu nutné identifikovat. Definice základních veličin je zobrazena na schématu níže:

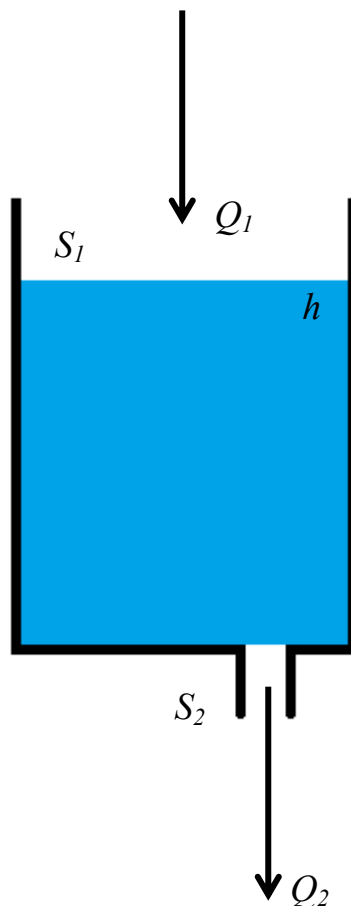
$Q_1$  – Přítok do nádrže. Velikost průtoku je měřena průtokoměrem a nastavována servoventilem. Průtokoměr nám dává hodnoty v  $\text{m}^3\text{h}^{-1}$

$Q_2$  – odtok z nádrže

$S_1$  – čtvercová plocha nádrže o stranách 23cm a 45cm

$S_2$  – Plocha kruhového odtoku z nádrže. Tuto plochu je možné měnit ručními ventily umístěnými na výstupních potrubí. Průměr potrubí a tudíž největší průměr odtoku je 4cm

$h$  – výška hladiny v nádrži. K jejímu měření slouží hladinoměr umístěný na dně nádrže.



Obr. 5 Schéma nádrže

Postup pro sestavení modelu pomocí matematicko-fyzikální analýzy je následující.

Rychlost odtoku kapaliny z nádrže je:

$$Q_2 = S_2 \cdot v_2 \text{ [m}^3\text{s}^{-1}\text{]} \quad (16)$$

Výtoková rychlost  $v_2$  není konstantní a závisí na množství kapaliny v nádrži, přesněji řečeno na hydrostatickém tlaku v místě odtokového otvoru. Pro určení je použit zákon o zachování energie.

Malá část kapaliny má na hladině nulovou rychlost a tudíž i nulovou kinetickou energii. Potenciální energie je:

$$E_{\text{pot}} = m \cdot g \cdot h \quad (17)$$

Při výtoku má ta samá část kapaliny energii kinetickou

$$E_{\text{kin}} = \frac{1}{2} \cdot m \cdot v_2^2 \quad (18)$$

Platí zákon o zachování energie

$$E_{\text{pot}} = E_{\text{kin}} \quad (19)$$



Tedy

$$m \cdot g \cdot h = \frac{1}{2} \cdot m \cdot v_2^2$$

Po úpravě

$$v_2 = \sqrt{2 \cdot g \cdot h}$$

Rychlost kapaliny  $v_2$  ve výtokovém otvoru se mění s druhou odmocninou výšky hladiny  $h$  v nádrži.

Objemový průtok odtoku je tedy

$$Q_2 = S_2 \cdot \sqrt{2 \cdot g \cdot h}$$

Objem kapaliny v nádrži je:

$$V = S_1 \cdot h \text{ [m}^3\text{]} \quad (20)$$

Rychlost jeho změny je

$$V'(t) = Q_1(t) - Q_2(t)$$

Tedy

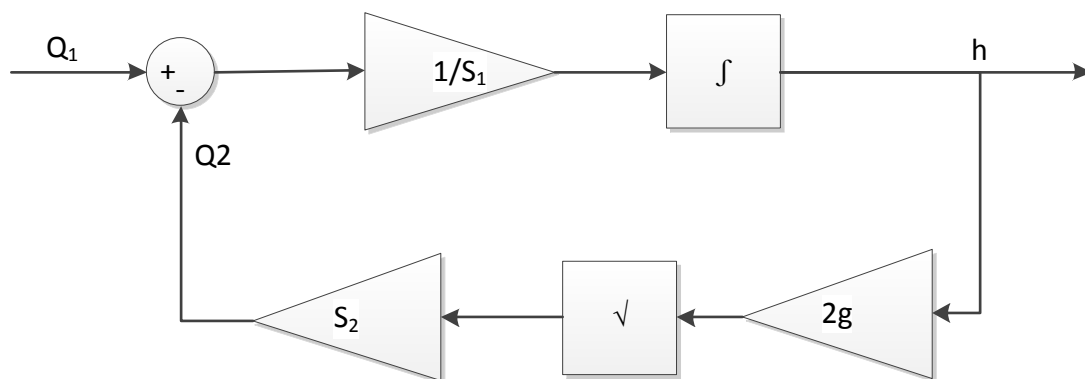
$$S_1 \cdot h'(t) = Q_1(t) - S_2 \cdot \sqrt{2 \cdot g \cdot h(t)}$$

Výsledná diferenciální rovnice

$$h'(t) = \frac{1}{S_1} \cdot [Q_1(t) - S_2 \cdot \sqrt{2 \cdot g \cdot h(t)}]$$

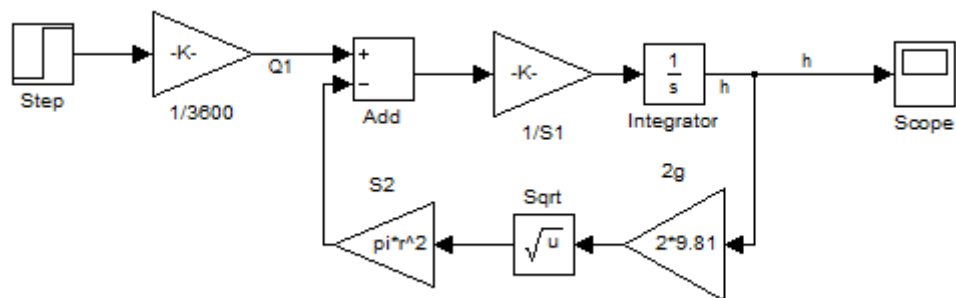
Jak je vidět, systém je nelineární, protože rychlost odtékání kapaliny se mění s druhou odmocninou výšky.

Model takového systému vypadá následovně

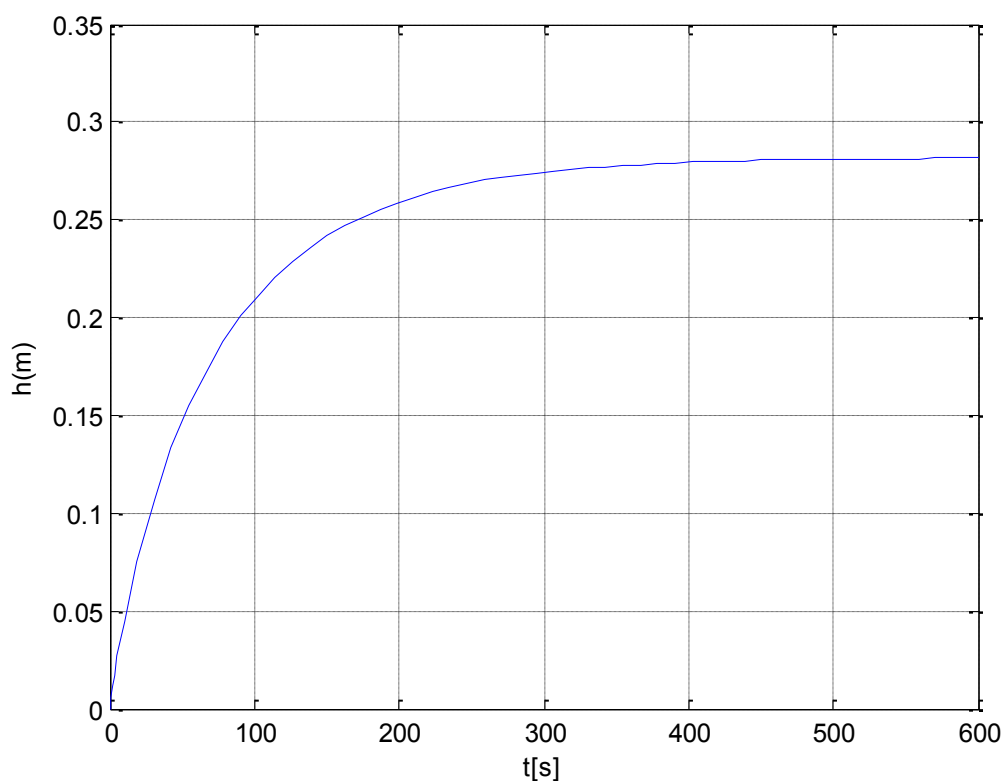


Obr. 6 Model nádrže s nuceným odtokem

## Model v Simulinku



Obr. 7 Model nádrže v Simulinku

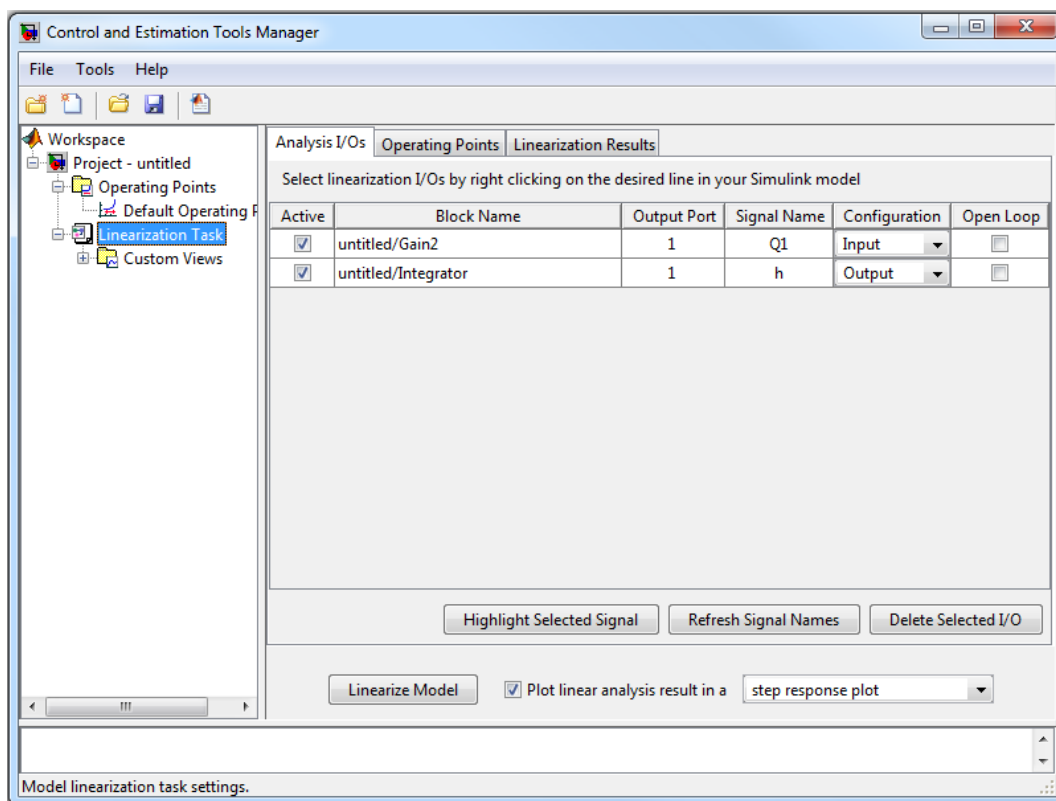


Obr. 8 Odezva na vstupní průtok  $2,4\text{m}^3/\text{h}$

## 4.2. Linearizace pomocí nástroje Linear Analysis

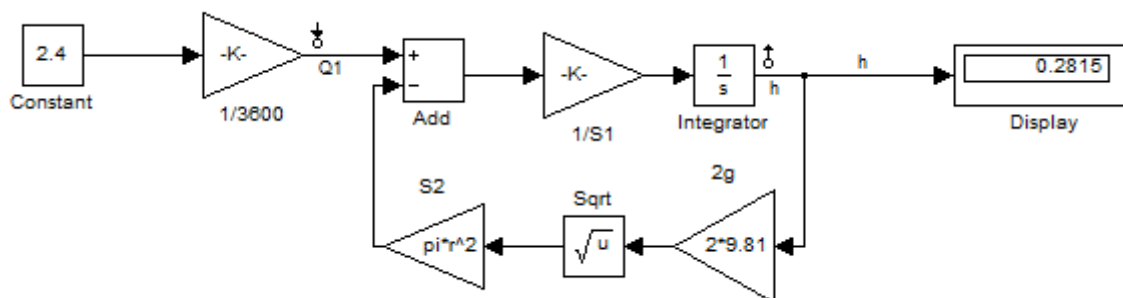
Linearizace slouží k vytvoření lineárního systému, jehož chování je velmi podobné chování původního nelineárního systému. Až na výjimky jsou téměř všechny dynamické procesy popsány diferenciálními rovnicemi nelineární. To způsobuje problémy, protože neexistují přesné analytické metody pro jejich řešení. Řešení lineárních diferenciálních rovnic je naproti tomu snadné. Proto je vhodné popsat nelineární dynamický systém lineárními diferenciálními rovnicemi. To je hlavní myšlenkou linearizace, avšak jedná se pouze o kompromis mezi jednoduchostí a přesností modelu.

Pro linearizaci nelineárních soustav obsahuje Matlab nástroj Linear Analysis. Ten jednoduše provede linearizaci v okolí zvoleného pracovního bodu.



Obr. 9 Nástroj Linear Analysis

Pro zjištění pracovního bodu slouží následující schéma



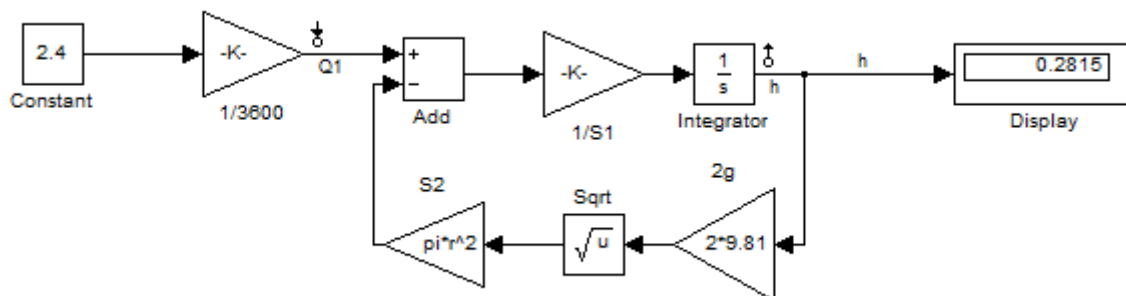
Obr. 10 Zjištění pracovního bodu pro linearizaci

Ze schématu lze vidět, že při objemovém průtoku  $Q_1 = 2,4 \text{ m}^3 \text{ h}^{-1}$  se výška hladiny ustálí na hodnotě  $h = 0,2815 \text{ m}$ . To je zvolený pracovní bod.

Po spuštění linearizace byla vypočítána nová lineární soustava 1. řádu s přenosem

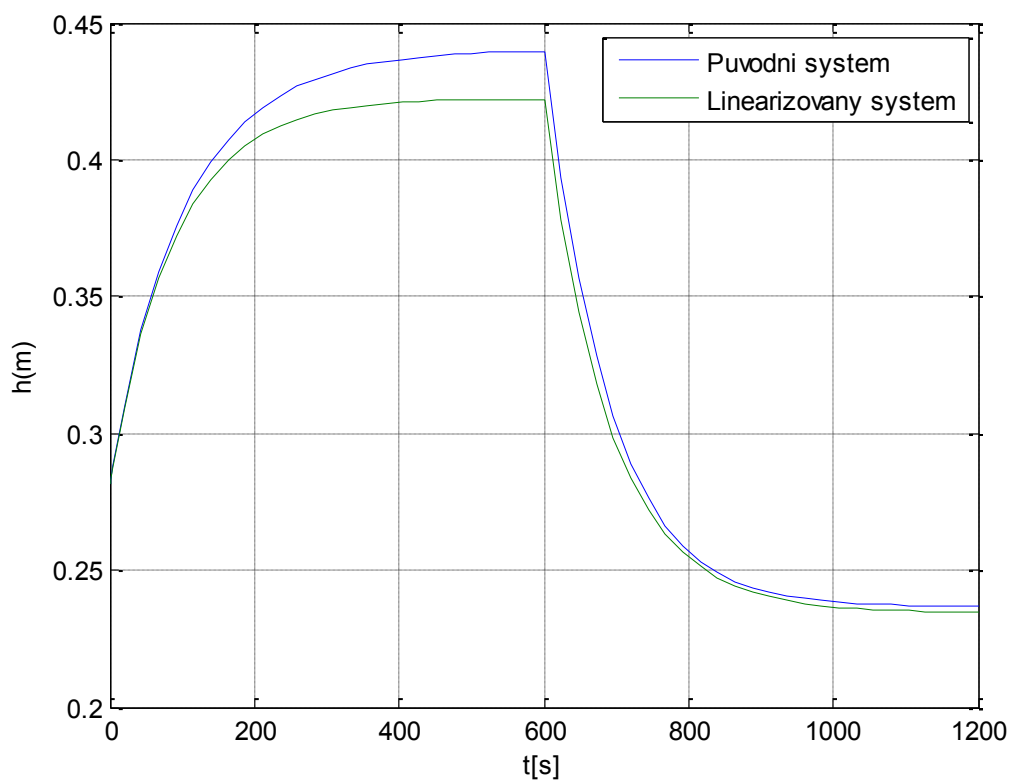
$$G(s) = \frac{9.456}{s + 0.01119}$$

Do nového modelu je nutné zahrnout také počáteční podmínky.



Obr. 11 Linearizovaný model

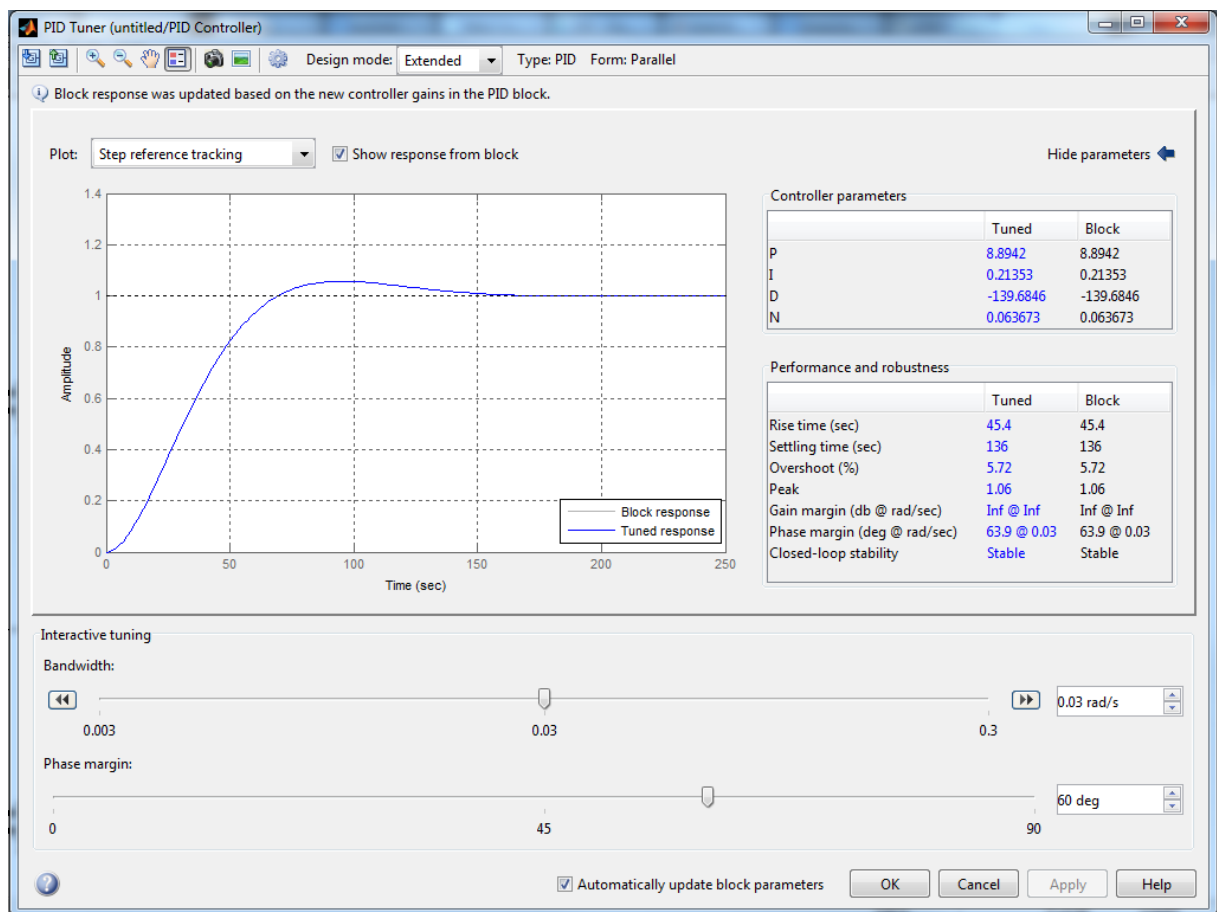
Odezva na změnu průtoku na  $3\text{m}^3/\text{h}$  a  $2,2\text{m}^3/\text{h}$  vypadá u obou modelů následovně



Obr. 12 Porovnání původního a linearizovaného systému

Návrh regulátoru lze provést jednoduše pomocí nástroje PID Tuner. Tento nástroj obsahuje přímo PID blok v Simulinku (od verze Matlabu 2010)

Blok PID regulátoru stačí pouze umístit do zpětnovazební smyčky a sám je schopen vypočítat konstanty zvoleného regulátoru.



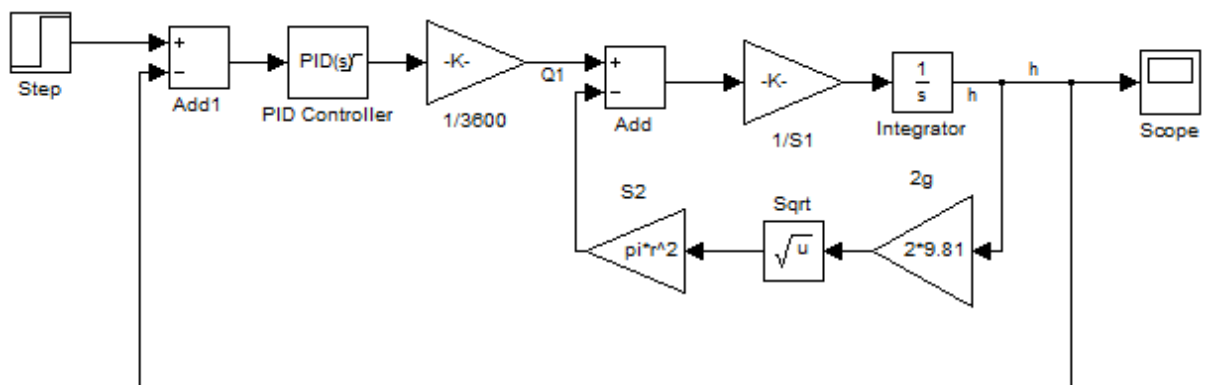
Obr. 13 Nástroj PID Tuner

Pro tuto soustavu byly experimentálně vypočteny konstanty.

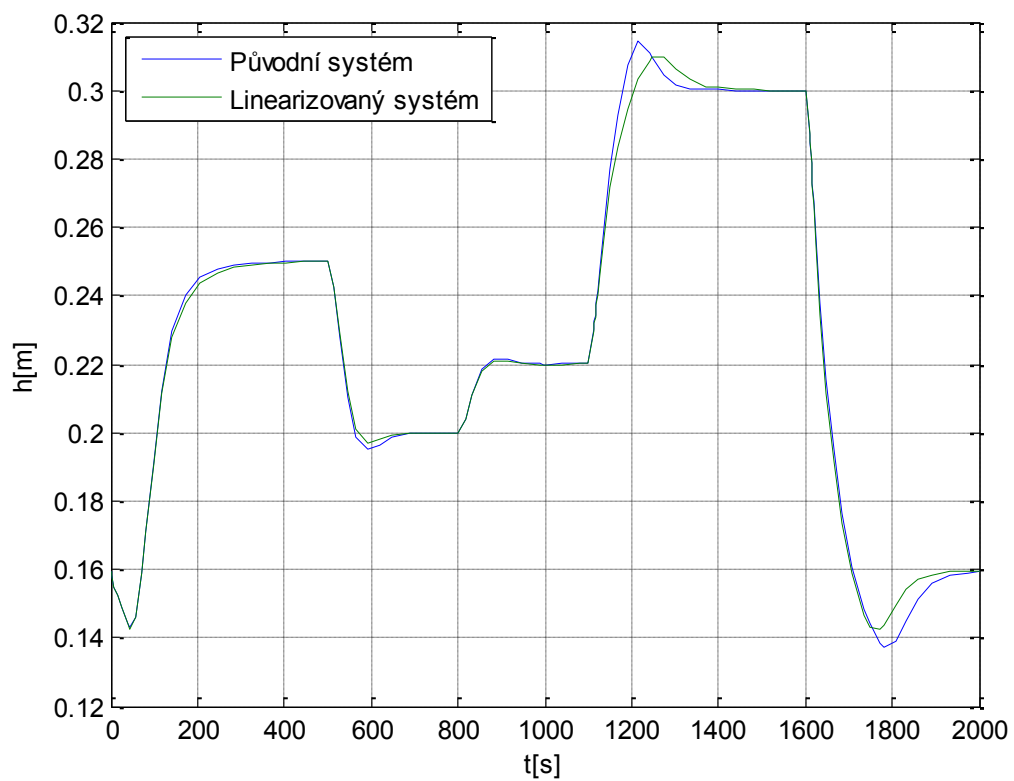
$$P = 8,89$$

$$I = 0,21$$

$$D = -139,68$$



Obr. 14 Zpětnovazební regulační obvod pro regulaci výšky hladiny



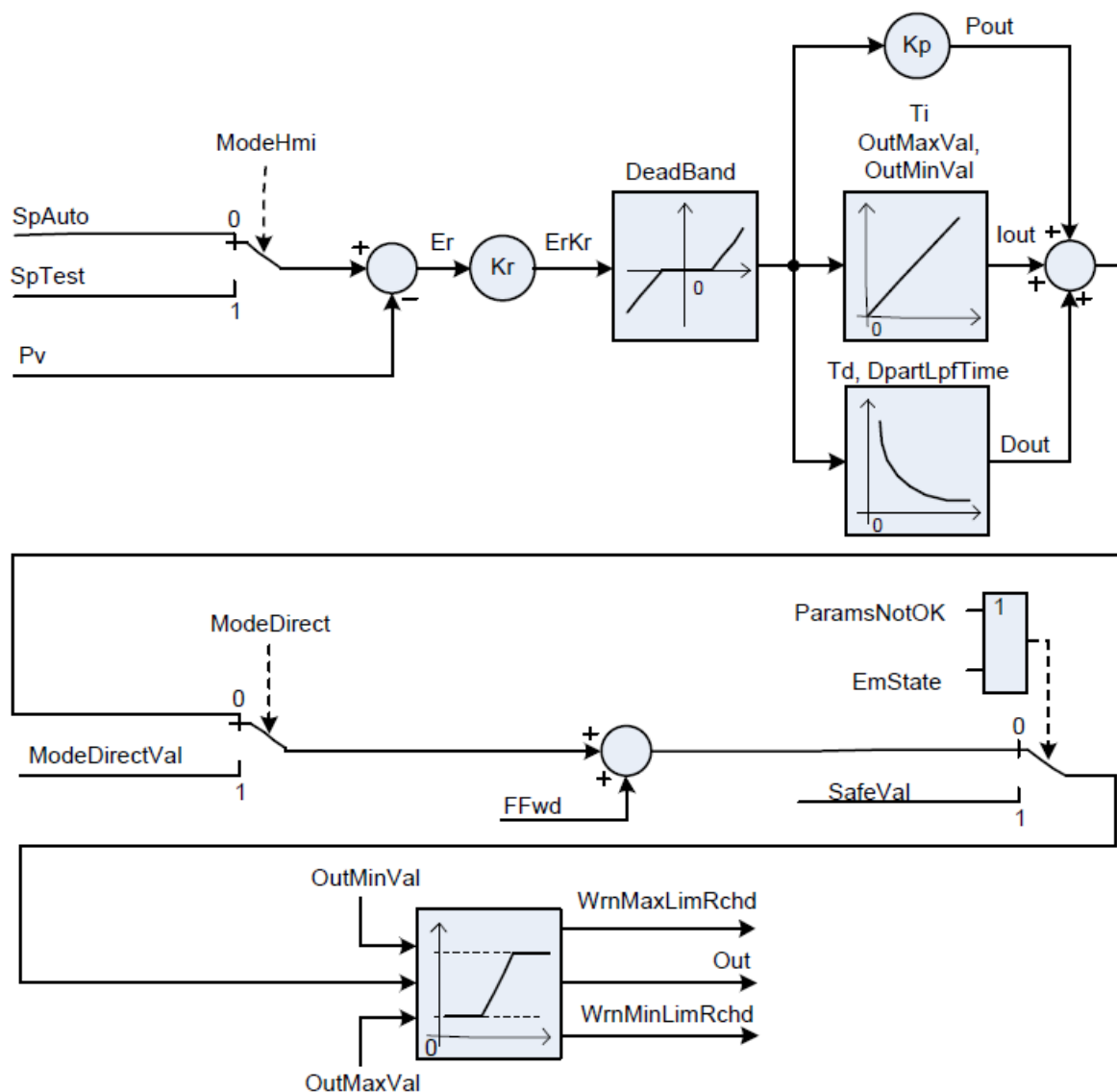
Obr. 15 Regulační děj

Na obrázku je vidět rozdíl regulace, mezi různými pracovními body. Pro regulaci na reálné soustavě byli použity konstanty, který byli nastaveny experimentálně vzhledem k odezvám regulace při změně požadované hodnoty výšky hladiny.



## 5. PID vyvinutý firmou INGETEAM

Firma Ingeteam má v prostředí STEP7 vyvinut vlastní PID regulátor, jehož schéma je uvedeno na obrázku níže.



Obr. 16 PID regulátor

Regulátor pracuje v několika módech. Obecně jsou to módy Safe, Direct, Auto a Test.

Mód Auto se zapíná bitem RqAuto z vizualizace. Žádaná hodnota je potom nastavována přímo ze vstupu regulátoru SpAuto.

Mód Test je volen bitem RqTest z vizualizace. Žádaná hodnota je nastavována vstupem SpTest z vizualizace. Aby bylo možno mód zapnout, musí být povolen bitem EnbModeTest ze vstupu regulátoru.

Mód Direct slouží k manuálnímu zadání akčního zásahu. Pokud je na vstupu ModeDirect log1, je na akční zásah přivedena hodnota ze vstupu ModeDirectVal. Tím je tedy vyřazen celý PID algoritmus.

Safe mód slouží jako pojistka, pokud dojde k chybě (např. špatným zadáním konstant). Poté je akční zásah nastaven na hodnotu ModeDirectVal ze vstupu regulátoru.

Nejvyšší prioritu má Safe mód. Dále sestupně Direct mód, Auto mód a nejnižší prioritu má Test mód.

Jednotlivé vstupy regulátoru jsou popsány v tabulce

Vstup	Typ	Význam
Ack	Bool	kvitace poruchy
OBStrtUp	Bool	první cyklus PLC
ModeDirect	Bool	zapnutí Direct módu
MaxLimRchEnbl	Bool	povolení varování o dosažení horního limitu
MinLimRchEnbl	Bool	povolení varování o dosažení dolního limitu
EnbModeTest	Bool	povolení Test módu
ModeDirectVal	Bool	hodnota akčního zásahu v Direct módu
SpAuto	Real	žádaná hodnota v módu Auto
Pv	Real	skutečná hodnota regulované soustavy
OutMinVal	Real	dolní limit akčního zásahu
OutMaxVal	Real	horní limit akčního zásahu
FFwd	Real	dopředná vazba

Tab. 2 Vstupy PID regulátoru

Jediný přímý výstup Out reprezentuje hodnotu akční zásah.

Vstup	Typ	Význam
RampOn	Bool	zapnutí rampy pro žádanou hodnotu
Kr	Real	zesílení regulační odchylky
Kp	Real	zesílení proporcionální složky
Ti	Real	integrační časová konstanta
Td	Real	derivační časová konstanta
SpTestSpd	Real	rychlost rampy pro žádanou hodnotu [jednotka/1s]
DbBnd	Real	mrtvé pásmo regulační odchylky
DpartLPFTime	Real	časová konstanta filtrace derivační složky
TrndRngMin	Real	minimální hodnota trendu ve vizualizaci
TrndRngMax	Real	maximální hodnota trendu ve vizualizaci

Tab. 3 Možnosti nastavení z vizualizace

Vstup	Typ	Význam
Ack	Bool	kvitace poruchy
RqTest	Bool	zapnutí Test módu
RqAuto	Bool	zapnutí Auto módu

Tab. 4 Příkazy z vizualizace

## 6. Popis školícího zařízení „Vodní svět“

### 6.1. Obecný popis

Zařízení „Vodní svět“ je určeno pro školící účely v oblasti řízení a ovládání. Po propojení s elektronickým řídicím systémem je možno regulovat výšku hladiny. Výška hladiny je regulována velikostí průtoku kapaliny s využitím ovládání čerpadla a servoventilu. Pro řízení napouštění a průtoku je instalován průtokoměr a dvě nádrže s hladinoměry. Zařízení také umožňuje sledovat změnu teploty a tlaku kapaliny při jeho ovládání. Ovládaným médiem je voda, kterou je možno směřovat pomocí osmi ručních ventilů a tím některé prvky z obvodu vyřadit.



Obr. 17 „Vodní svět“

Hlavní nosná část je vyrobena z plastu. Na tuto část jsou namontovány potrubní trasy s elektrickými a elektronickými prvky. Hlavní část se skládá z hlavního panelu, záchytné jímky rozdělené do dvou sekcí. Nad touto jímkou je menší přečerpávací jímka s přepadem. Na boku zařízení je čerpadlo, které do potrubních rozvodů na hlavním panelu dodává vodu. Čerpadlo je možné regulovat přes frekvenční měnič. Dále je v obvodu servoventil, průtočný ohřívač a pro sledování stavu ovládaného média je v obvodu umístěn průtokoměr, teplotní čidla, tlaková čidla a čidla pro měření výšky hladiny. Výstupy těchto čidel jsou svedeny do vzdálené periferie ET 200S, která je k CPU, kde se výstupy z těchto čidel vyhodnocují připojena pomocí komunikační sběrnice PROFIBUS – DP.

### Technické parametry školícího zařízení

Vnitřní rozměry nádrže	1976 x 488 x 685mm
Celkové rozměry	2521 x 692 x 1957mm
Užitný objem	0,66m <sup>3</sup>
Materiál zařízení	PP-H
Hmotnost zařízení	174,6kg
Celková hmotnost	834,6kg
Náplň	Voda
Max teplota lázně	60°C
Čerpadlo	Pedrollo CP 100
Datum výroby	2011

Tab. 5 Technické parametry školícího zařízení

### 6.2. Snímače relativního tlaku s keramickou membránou D2411

Na zařízení jsou nainstalovány D2411 snímače relativního tlaku s keramickou membránou. Hlavní oblastí použití snímačů je chemický a farmaceutický průmysl, potravinářský průmysl, zdravotní technika, zkušební zařízení a obecná měření tlaku. Snímače převádějí tlak plynu nebo kapalin na elektrický signál. Jsou vhodné jak pro statická, tak dynamická měření. Výstupní signál snímače je teplotně kompenzován pomocí vestavěné elektroniky, zesílen a normován. Nula snímače odpovídá atmosférickému tlaku.[5]



Obr. 18 Snímač tlaku D2411

Snímač měří tlak v rozsahu 1Bar až 400Bar při volitelném proudovém, nebo napěťovém výstupním signálu 4 až 10mA nebo 0 až 10V s přesností měření 0,5%.

### 6.3. Snímače tlaku s keramickou čelní membránou LMK 351

Snímače LMK 351 jsou určeny pro měření tlaků a výšky hladiny kapalin, emulzí a kalů ve speciálních technologických procesech. Použití snímačů je v chemickém průmyslu, lékařské technice, farmaceutickém a papírenském průmyslu. Tlaku úměrný výstupní signál je následně teplotně kompenzován a normován.[6]



Obr. 19 Snímač tlaku LMK351

Snímač měří tlak v rozsahu 0,04Bar až 60Bar při volitelném proudovém, nebo napěťovém výstupním signálu 4 až 10mA nebo 0 až 10V s přesností měření 0,35%.

#### 6.4. Indukční průtokoměr FLONET FN 20.1

Indukční průtokoměr FLONET FN 20.1 je měřidlo objemového průtoku vodivých kapalin v uzavřeném potrubí. Umožňuje obousměrné měření průtoku v pásmu rychlostí proudění 0,1 - 10 m/s. Průtokoměr je založen na Faradayově indukčním zákonu. Čidlo se skládá z nemagnetické nevodivé trubky, ve které jsou kolmo na směr magnetických siločar zabudovány dvě měřicí elektrody pro snímání indukovaného napětí. Pro vytvoření střídavého magnetického pole jsou na trubce umístěny dvě cívky rovnoběžné s rovinou snímacích elektrod. Pohybem vodivé tekutiny vzniká na měřicích elektrodách indukované napětí, které je přímo úměrné rychlosti proudění.[7]

Průtokoměr se skládá ze dvou částí, z čidla, kterým protéká měřená kapalina a z vyhodnocovacího zařízení, které převádí signál na nízké úrovni na unifikovaný signál vhodný ke zpracování průmyslovými přístroji.[7]



Obr. 20 Indukční průtokoměr

K zobrazení měřené hodnoty slouží dvouřádkový alfanumerický displej, na kterém společně s klávesnicí lze měnit některé provozní parametry měřidla. Průtokoměr obsahuje dva pasivní programovatelné digitální výstupy a jeden proudový aktivní výstup, jehož signál je přímo úměrný

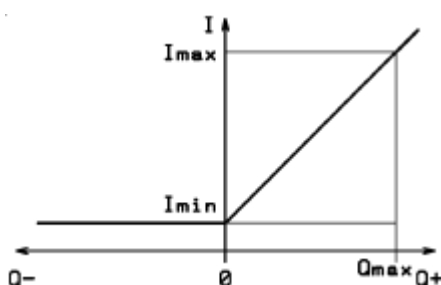
objemovému průtoku. Pro digitální komunikaci je vybaven také komunikačním rozhraním RS485. Pokud je přístroj použit jako fakturační měřidlo, mohou být některé funkce zablokovány tak, aby uživatel nemohl ovlivnit jeho přesnost.

Použití průtokoměru je omezeno pouze požadavkem, že měřená kapalina musí být nemagnetická a elektricky vodivá s minimálním požadavkem vodivosti  $20\mu\text{S/cm}$ . [7]

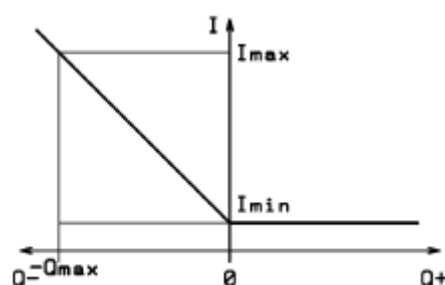
### **Programovatelný proudový výstup**

Zařízení má k dispozici aktivní programovatelný proudový výstup, který je galvanicky oddělen od ostatních částí průtokoměru. Pracuje do zátěže  $1000\Omega$ . V závislosti na průtoku může pracovat ve čtyřech režimech a ve dvou přepínatelných rozsazích 0 až 20mA a 4 až 20mA. [7]

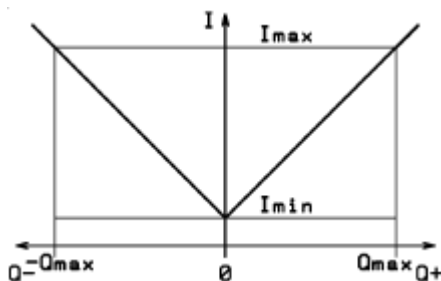
Režimy proudového výstupu jsou znázorněny na obrázcích níže. Přepínání mezi režimy je realizováno pomocí klávesnice a alfanumerického displeje.



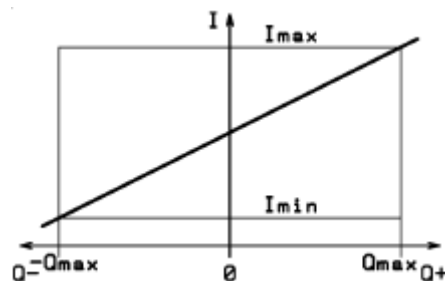
Obr. 21 Proud pro  $0 \gg +Q$



Obr. 22 Proud pro  $0 \gg -Q$



Obr. 23 Proud pro  $|Q|$



Obr. 24 Proud pro  $-Q \gg +Q$  [7]

Analogový výstup je řízen 12-ti bitovým DA převodníkem. Rozsah 0 až 20mA je tedy rozdělen do 4096 kroků, jeden krok tedy představuje asi 0,005mA. Rozsah 4 až 20mA je vytvořen pouze softwarovým snížením počtu kroků převodníku. Rozlišení je tedy stejné. [7]

Mimo uvedené režimy je analogový výstup schopen pracovat také v režimu pevného proudu a může tak sloužit jako zdroj pro pasivní digitální výstupy.

### **Digitální multifunkční výstupy**

Průtokoměr je vybaven dvěma binárními multifunkčními výstupy galvanicky oddělenými. Výstupy jsou pasivní a ke své činnosti potřebují vnější zdroj o výstupním proudu 1 až 50mA při trvalém sepnutí. [7]

Lze zvolit mezi těmito režimy:



### **Pulsní Výstupy**

Na výstup je generován puls bezprostředně po přetečení předvoleného objemu.

### **Frekvenční výstupy**

Na výstup je generována frekvence v rozsahu 1 až 10000Hz přímo úměrná průtoku kapaliny. Elektronika disponuje pouze jedním generátorem frekvence, není proto možné nastavit různou frekvenci na každý výstup.

### **Záporný průtok**

Při záporném průtoku je výstup sepnut, při nezáporném rozepnut.

### **Porucha**

V případě vzniku poruchy je výstup sepnut po celou dobu trvání poruchy.

### **Překročení nebo pokles mezní hodnoty**

V případě překročení nebo poklesu přednastavené mezní hodnoty je výstup sepnut nebo rozepnut.

### **Čištění elektrod**

Výstup je sepnut, pokud probíhá čištění elektrod.

## **6.5. Ventil RV103 s elektrickým pohonem SQX62**

SQX62 je elektrický pohon pro ovládání přímých a trojcestných ventilů se jmenovitým zdvihem 20mm. Pohon je řízen signálem 4 až 20mA, 0 až 10V nebo 0 až 1000Ω. Tento signál řídí motor elektronickým mikroprocesorem. Motor přes převodový mechanismus generuje požadovaný zdvih.[8]



Obr. 25 Pohon SQX62

Pohon je také vybaven analogovým výstupem pro sledování polohy ventilu. Výstupní rozsah je možno zvolit z možností 0 až 10V, 0 až 1000Ω nebo 4 až 20mA

Protože má použitý ventil RV103 poloviční jmenovitý zdvih, než pohon SQX62, je nutno nastavení polohy a měření polohy také přepočíst na rozsah 0 -10V, 500 až 1000 Ω nebo 12 až 20mA. To lze zajistit přepočtem jednotek AD a DA převodníku PLC.

$$PQW512 = \frac{nastavenaPoloha}{2} + 13824$$

$$skutecnaPoloha = \frac{PIW522 - 13824}{2}$$

### **Technické parametry pohonu**

Napájecí napětí	24V
Příkon	6,5VA
Ovládací signály	0 - 10V, 4 - 20mA, 0 - 1000Ω
Doba přechodu	35s
Jmenovitá síla	700N
Zdvih	20mm
Max teplota média	140°C
Teplota okolí	minus 15 až 50°C
Hmotnost	1,6kg

Tab. 6 Parametry pohonu SQX62

## **7. Linearizační funkce pro potlačení nelinearity proporcionálního ventilu**

Interpolace a aproximace funkcí nebo experimentálních dat zahrnuje řadu technik. Při výběru vhodné aproximace postupujeme tak, že předem zvolíme tvar aproximující funkce, ve které vystupují nějaké proměnné parametry, a hodnoty těchto parametrů se pak snažíme určit tak, aby získaná aproximace vyhovovala našim požadavkům.

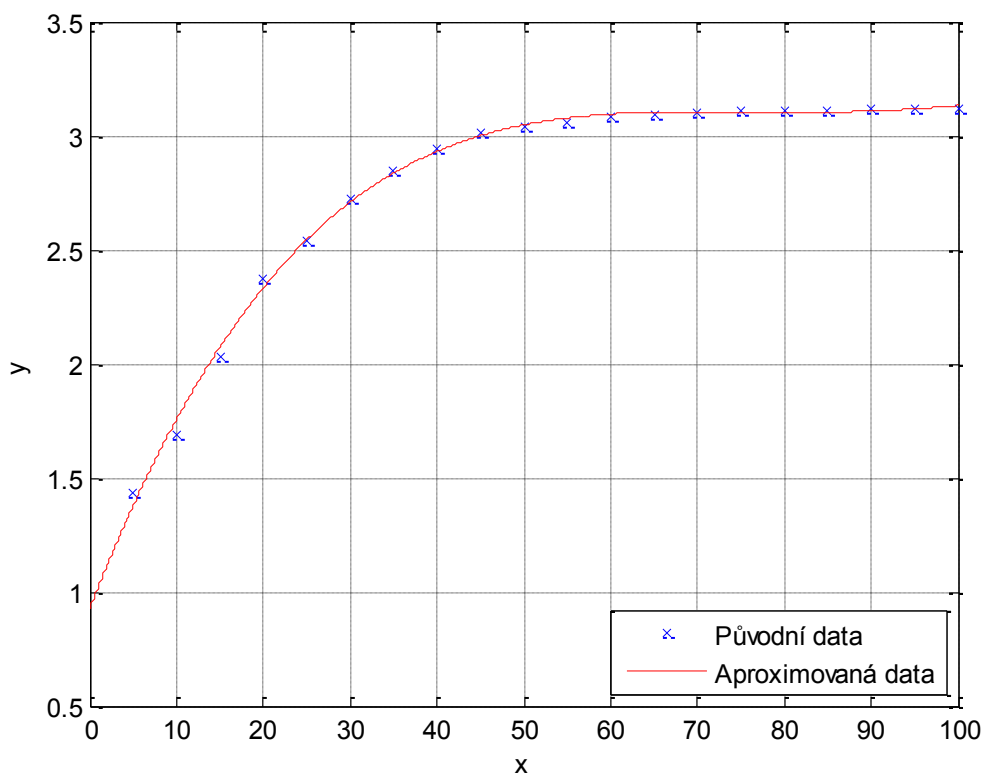
### **7.1. Interpolace funkce**

#### **Polynomická interpolace**

Mezi nejznámější postupy patří polynomická interpolace, která hledá polynom nejmenšího možného stupně. Ukázka výpočtu polynomické interpolace v programu Matlab je uvedena níže.

```
1. y1 = polyfit(y, x ,4);
2. y_aprox=polyval(y1, 0:100);
```

Na obrázku je vidět srovnání aproximační funkce s polynomem čtvrtého řádu s původními naměřenými daty.



Obr. 26 Polynomičká interpolace

Chceme-li interpolovat funkci, která je dána přesně svými hodnotami a požadujeme-li, aby aproximace procházela zadanými body, použijeme aproximaci interpolačním polynomem. Aproximace nám potom slouží k získání přibližné hodnoty zadané funkce v libovolném bodě intervalu. Máme-li zadány hodnoty funkce  $f$  v  $n + 1$  různých bodech, tzn., máme zadáno  $n + 1$  tzv. interpolačních podmínek pro polynom, je zřejmé, že stupeň hledaného polynomu bude  $n$  (polynom  $n$ -tého stupně má  $n + 1$  koeficientů). Lze ukázat, že mezi všemi polynomy nejvýše  $n$ -tého stupně existuje právě jeden, který je interpolačním polynomem pro zadanou funkci. Pro určení interpolačního polynomu existuje několik postupů, ale pro zadanou funkci všechny postupy určí stejný polynom.[9]

### **Lagrangeův interpolační polynom**

Časté je použití tzv. Lagrangeova interpolačního polynomu. Ten je ve tvaru:[9]

$$L_n(x) = \sum_{i=0}^n f(x_i) \cdot l_i(x) \quad (21)$$

Kde  $l_i$  jsou polynomy  $n$ -tého stupně, pro které platí

$$l_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (22)$$

Zde je uveden postup pro výpočet Lagrangeova interpolačního polynomu třetího stupně pomocí programu Matlab.

```
x=[10 40 70 100]
y=[1.73 3.16 3.38 3.42]

for i=1:length(x)

    A(i,:)=x(i)-x;

end

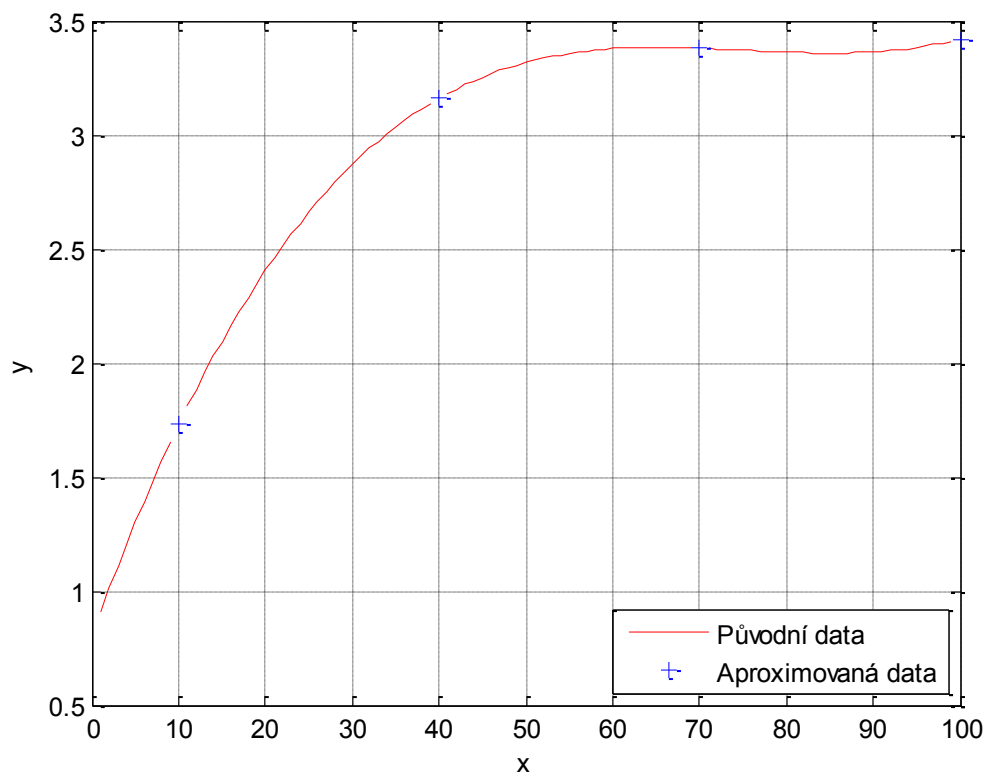
for xv = 1:100

    for j=1:i A(j,j)=xv-x(j); end

    D=prod(A')
    S=prod(diag(A))
    L(xv)=S*sum(y./D);

end
```

Na výsledku je vidět, jak křivka Lagrangeova interpolačního polynomu prochází přesně zadanými body.



Obr. 27 Lagrangeův interpolační polynom

## **Newtonův interpolační polynom**

Nevýhodou Lagrangeova interpolačního polynomu je nutnost opětovného přepočítání všech členů při přidání dalšího bodu. Z tohoto hlediska bývá efektivnější využití Newtonova interpolačního polynomu.[9]

$$N_n(x) = a_0 + a_1 \cdot (x - x_0) + a_2 \cdot (x - x_0) \cdot (x - x_1) + \dots + a_n \cdot (x - x_0) \cdot (x - x_1) \dots (x - x_{n-1}) \quad (23)$$

Koeficienty  $a_i$  lze dopočítat pomocí tzv. proměnných diferencí.

Výhodou Newtonova interpolačního polynomu je, že přidáme-li další bod interpolace  $x_{n+1}$ , nemusíme celý výpočet opakovat, ale postačí dopočítat pouze koeficient  $a_{n+1}$  a přidat ho k ostatním koeficientům. Výpočet vede ke stejnému výsledku jako pomocí Lagrangeova interpolačního polynomu.[9]

Interpolační polynomy vyšších řádů není vhodné používat pro aproximaci hodnot funkce mimo interval, který obsahuje uzly interpolace, protože absolutní hodnota polynomu nabývá velkých hodnot. Obecně není vhodné interpolovat polynomem funkci, která je dána velkým počtem svých hodnot. Stupeň interpolačního polynomu je potom velký a vlastnosti polynomu vysokého stupně nejsou dobré nejen mimo interval obsahující uzly interpolace, ale také uvnitř. Interpolační polynom může směrem ke krajním bodům vlivem nepřesností ve vstupních datech výrazně oscilovat. Tyto oscilace lze minimalizovat použitím neekvidistantních uzlů. Interpolace polynomem není obecně vhodná pro funkce, které mají asymptotu. Interpolace polynomem také není vhodná pro funkce, které jsou dány tabulkou a jsou zatíženy velkou chybou.[9]

## **Lineární interpolace**

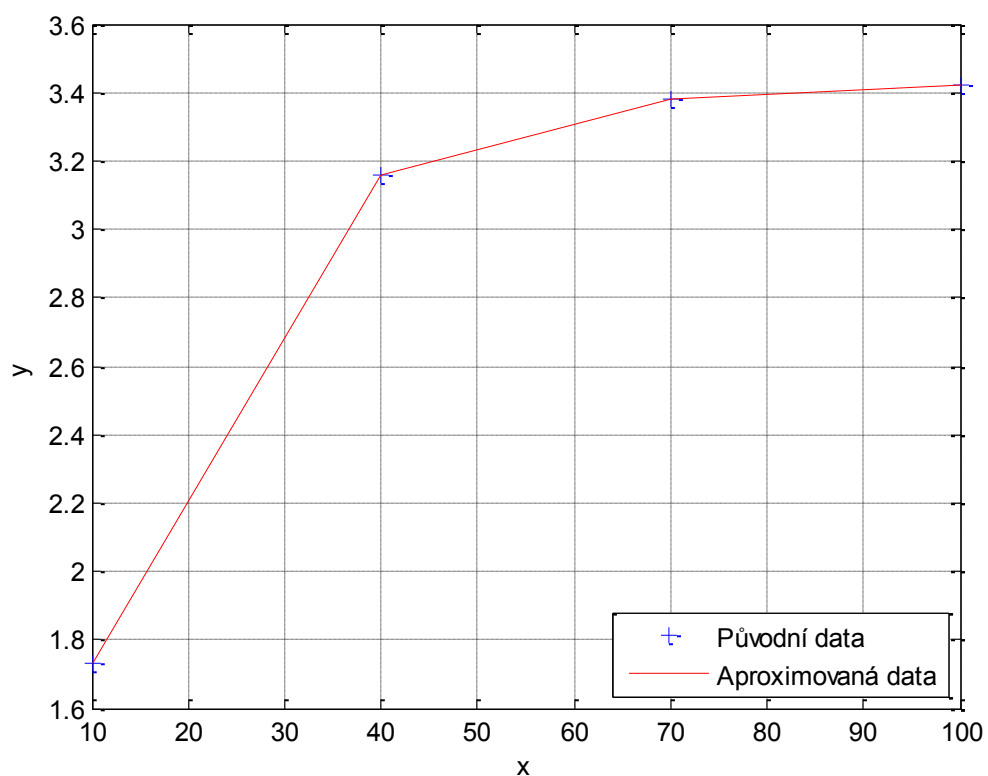
Další možností je použití lineární interpolace. Lineární interpolace je metoda prokládání křivek za použití lineárních mnohočlenů. Pokud známe dva body, které jsou dány souřadnicemi  $(x_0, y_0)$  a  $(x_1, y_1)$ , tak lineární interpolací rozumíme přímku mezi těmito body. Hledaná hodnota  $y$ , která se nachází na přímce mezi těmito body, je dána rovnicí

$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}$$

Úprava této rovnice, ve které je neznámou  $y$  v rovnici pro  $x$  dostaneme:

$$y = y_0 + (x - x_0) \cdot \frac{y_1 - y_0}{x_1 - x_0}$$

Lineární interpolace je snadná cesta k zaplnění mezer v tabulce. Její výpočet je jednoduchý a při dostatečném počtu prvků bývá poměrně přesný. Proto bývá často používána například v počítačové grafice a její operace jsou vestavěny v hardwaru všech moderních počítačových grafických karet. Často jsou používány jako stavební blok pro více komplexnější operace, například bilineární interpolace, která bývá zhotovena ve dvou lineárních interpolacích.

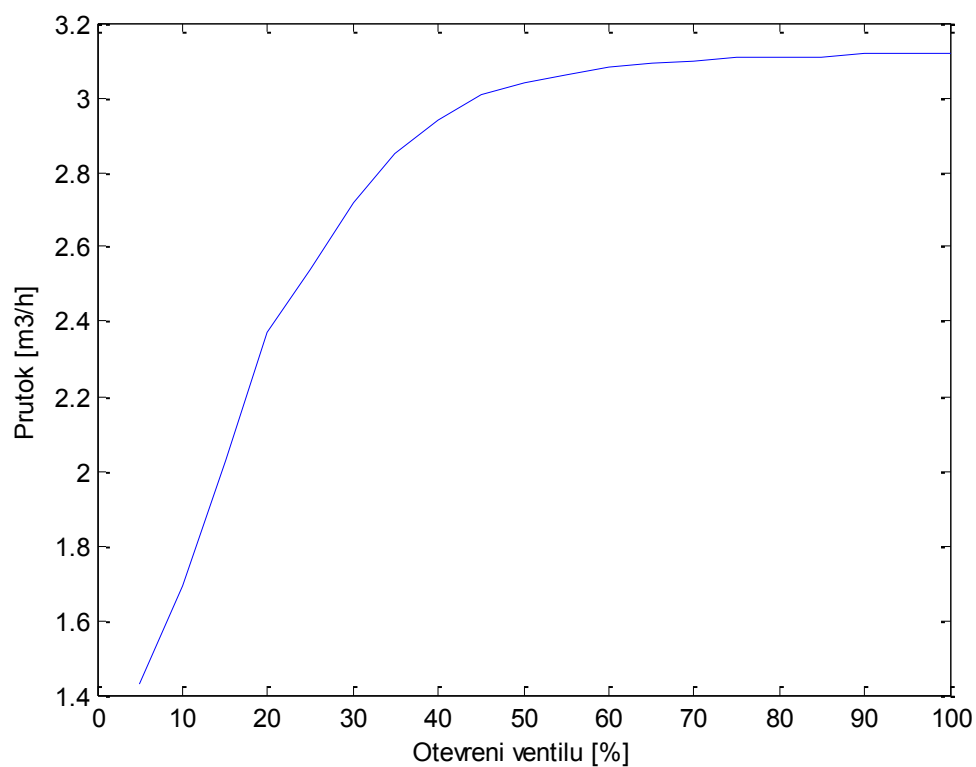


Obr. 28 Lineární interpolace

## 7.2. Naměřená data

Změřená průtočná charakteristika je na obrázku. Z charakteristiky je patrná velká nelinearita ventilu. Průtok se efektivně mění v rozmezí 0 až 50%. Od této hodnoty se průtok mění pouze minimálně a zůstává víceméně konstantní.





Obr. 29 Průtočná charakteristika proporcionálního ventilu

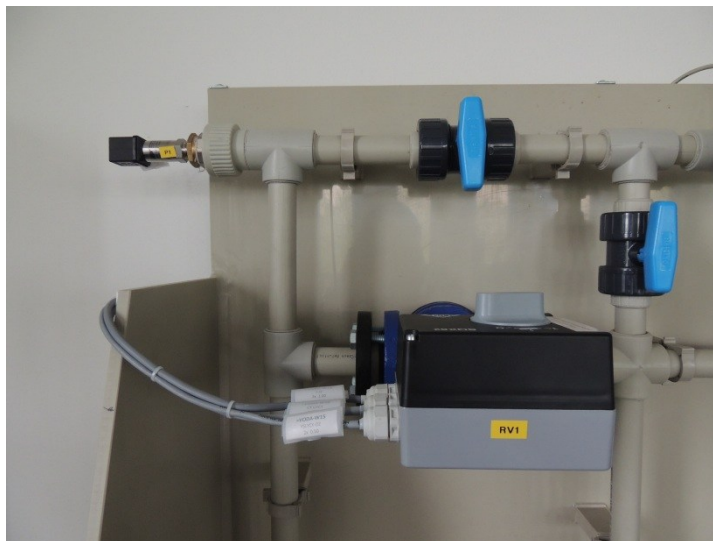
Otevření ventilu [%]	Průtok [m3/h]
100,00	3,14
95,00	3,12
90,00	3,12
85,00	3,11
80,00	3,11
75,00	3,11
70,00	3,10
65,00	3,09
60,00	3,08
55,00	3,06
50,00	3,04
45,00	3,01
40,00	2,94
35,00	2,85
30,00	2,72
25,00	2,54
20,00	2,37
15,00	2,03
10,00	1,69
5,00	1,43

Tab. 7 Tabulka naměřených hodnot průtoku v závislosti na otevření ventilu

Regulace průtoku s ventilem s takovou charakteristikou by byla velmi nekvalitní, proto bylo nutné vytvořit funkci, která na základě naměřených hodnot lineární interpolací vypočítává otevření ventilu pro žádaný průtok.

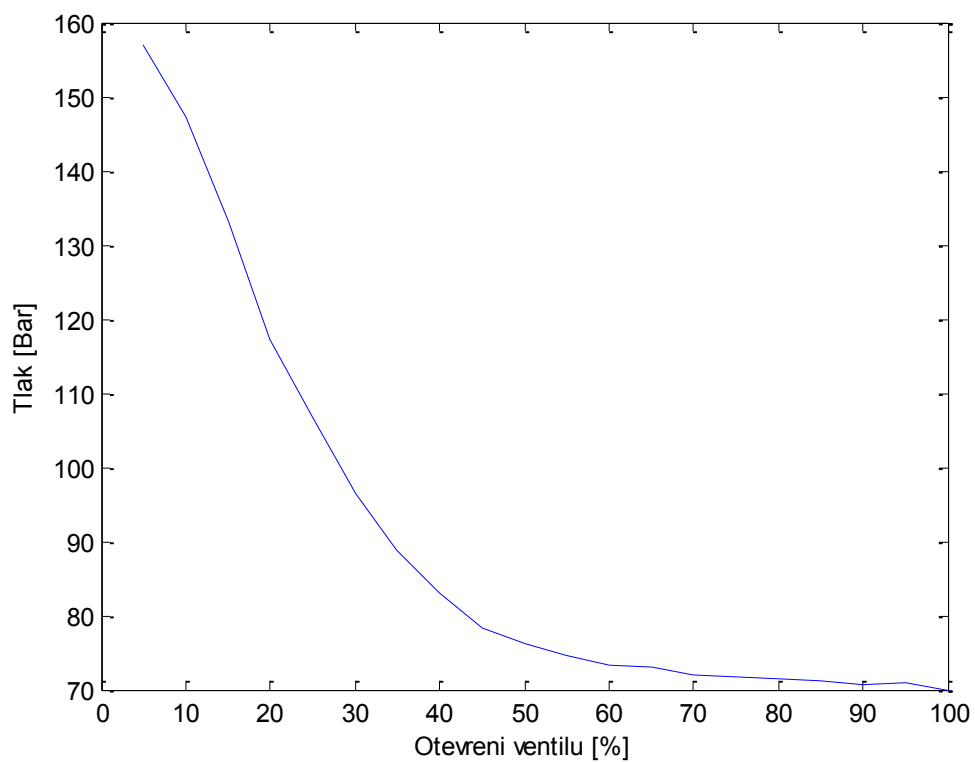
Požadavkem na funkci bylo, aby dokázala reagovat na změnu tlaku před ventilem. Ta může nastat například změnou frekvence otáčení čerpadla.

Pro sledování změny vstupního tlaku byl použit tlakoměr *PI* na vstupní straně ventilu.



Obr. 30 Ventil a tlakoměr pro měření tlaku před ventilem

Změna tlaku na vstupu je také nelineární. Stejně jako u ventilu se tlak mění efektivně v rozmezí 0 až 50% otevření, poté zůstává konstantní. Tlak na rozdíl od průtoku s rostoucím otevřením klesá.



Obr. 31 Charakteristika tlaku na vstupu ventilu

Otevření ventilu [%]	Tlak [Bar]
100,00	3,14
95,00	3,12
90,00	3,12
85,00	3,11
80,00	3,11
75,00	3,11
70,00	3,10
65,00	3,09
60,00	3,08
55,00	3,06
50,00	3,04
45,00	3,01
40,00	2,94
35,00	2,85
30,00	2,72
25,00	2,54
20,00	2,37
15,00	2,03
10,00	1,69
5,00	1,43

Tab. 8 Tabulka naměřených hodnot vstupního tlaku v závislosti na změně otevření

Protože ani vstupní tlak není lineární, musí funkce odstraňovat i tuto nelinearitu.

### 7.3. Vstupní data pro linearizační funkci

Pro potřeby funkce bylo zapotřebí naměřit hodnoty, pomocí kterých by funkce dopočítávala výstup. Protože nebylo možno změřit průtočné charakteristiky pro konstantní tlak na vstupu, musely být naměřeny charakteristiky při konstantním otevření a změně tlaku na vstupu. Ke změně tlaku byl použit ruční ventil umístěný za čerpadlem



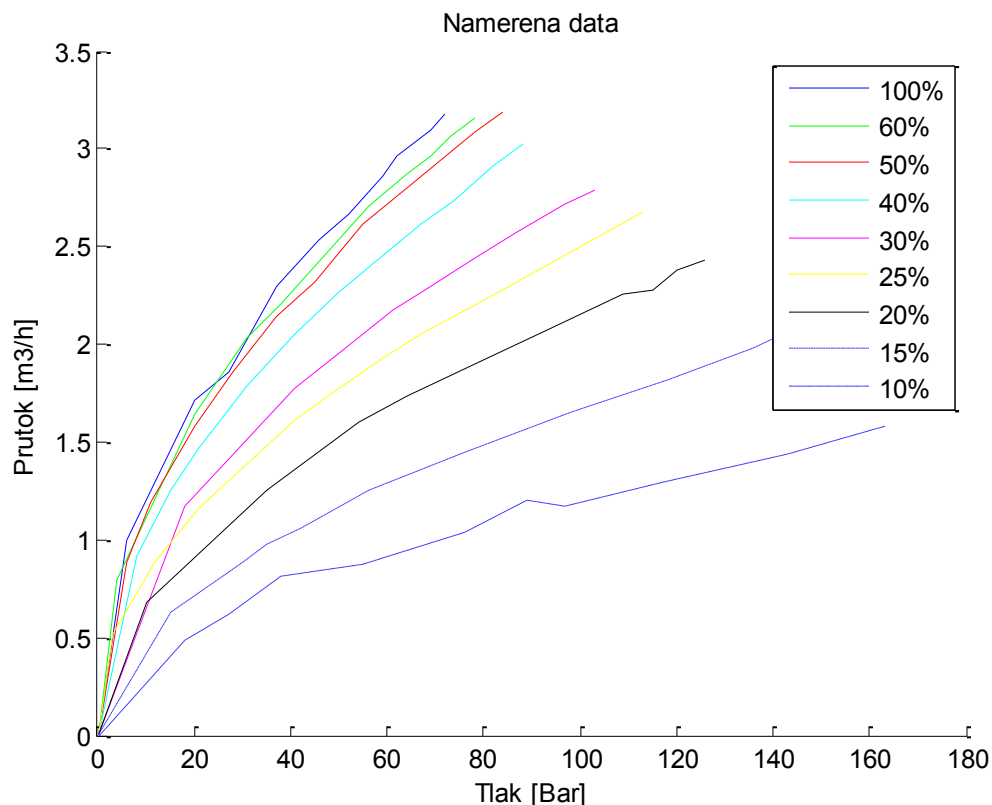
Obr. 32 Ruční ventil pro změnu tlaku před proporčním ventilem

Měření bylo provedeno při otevření 100%, 60%, 50%, 40%, 30%, 25%, 20%, 15%, 10% a 5%. Postupně byl ručním ventilem snižován tlak a průtokoměrem odečítán průtok. Velikosti otevření byly vzhledem k charakteristice zvoleny v daných rozmezích tak, aby byly data rovnoměrně rozprostřeny.

Naměřená data jsou uvedena v tabulkách na další stránce. Hodnoty tlaku jsou uvedeny v Bar a hodnoty průtoku jsou uvedeny v  $\text{m}^3/\text{h}$ .

Namerena data					
Otevreni 100%		Otevreni 60%		Otevreni 50%	
Tlak	Prutok	Tlak	Prutok	Tlak	Prutok
72	3,18	78	3,16	84	3,19
69	3,1	75	3,1	78	3,09
62	2,96	73	3,07	64	2,8
59	2,86	69	2,96	55	2,61
52	2,67	64	2,87	45	2,32
46	2,53	56	2,71	37	2,14
37	2,3	38	2,21	28	1,86
27	1,86	31	2,03	20	1,58
20	1,71	20	1,64	11	1,19
6	1	4	0,79	6	0,89
0	0	0	0	0	0
Otevreni 40%		Otevreni 30%		Otevreni 25%	
Tlak	Prutok	Tlak	Prutok	Tlak	Prutok
88	3,02	103	2,79	113	2,68
82	2,91	97	2,72	101	2,51
74	2,74	87	2,57	83	2,27
67	2,61	78	2,44	67	2,05
50	2,27	68	2,28	59	1,93
41	2,05	61	2,17	47	1,72
31	1,79	51	1,97	41	1,61
21	1,47	41	1,78	21	1,16
15	1,25	35	1,62	12	0,89
8	0,92	18	1,17	3	0,52
0	0	0	0	0	0
Otevreni 20%		Otevreni 15%		Otevreni 10%	
Tlak	Prutok	Tlak	Prutok	Tlak	Prutok
126	2,43	143	2,06	163	1,58
120	2,38	136	1,98	143	1,44
115	2,28	118	1,82	117	1,29
109	2,26	98	1,65	97	1,17
93	2,07	76	1,45	89	1,2
79	1,91	56	1,25	76	1,04
64	1,73	42	1,06	55	0,87
54	1,6	35	0,98	38	0,81
35	1,25	30	0,89	27	0,62
10	0,68	15	0,63	18	0,49

Tab. 9 Naměřená data



Obr. 33 Vykreslení naměřených dat

Přesné nastavení tlaku tak, aby bylo možno porovnat hodnoty průtoku při různém otevření nebylo možné, proto byly následně naměřené hodnoty aproximovány polynomem druhého řádu. Pro přesnější aproximaci byl odstraněn bod (0;0) a protože při vyšších hodnotách otevření nebylo možno naměřit průtok pro vyšší hodnoty tlaku, musely být tyto hodnoty odhadnuty. Odhad byl proveden lineárním prodloužením posledních dvou hodnot a velikost průtoku byla odečtena pro tlak 150Barů. Následnou aproximací byl získán průtok pro velikost vstupního tlaku 0Bar, 25Bar, 50Bar, 75Bar, 100Bar a 150Bar při jmenovitém otevření ventilu. Pro všechny výpočty byl použit program Matlab.

Ukázka části kódu s úpravami je níže.

```
3. V_100(length(V_100), :) = [];
4. pt100 = 150;
5. aux100 = V_100(1:2,:);
6. aux100_apx = polyfit(aux100(:,1), aux100(:,2) ,1);
7. aux100_aprox=polyval(aux100_apx, pt100);
8. aux100 = [pt100 aux100_aprox];
9. V_100 = [aux100; V_100];
```

Samotná aproximace polynomem druhého řádu.

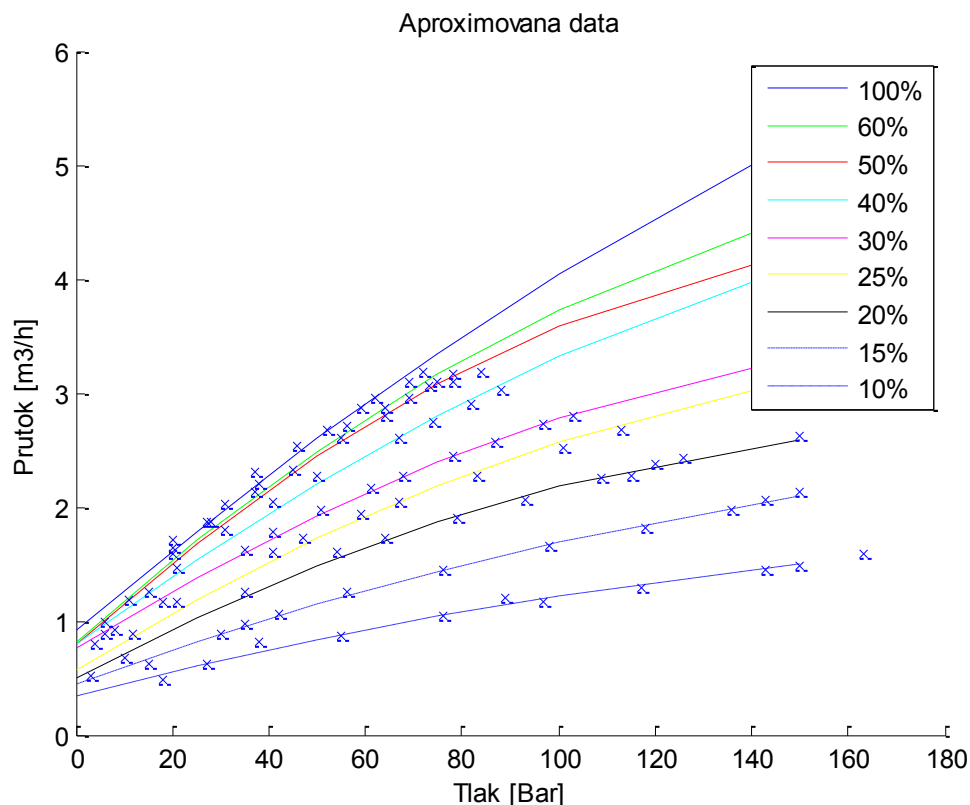
```
1. V_100_apx=polyfit(V_100(:,1), V_100(:,2) ,rad);
2. x=[0; 25; 50; 75; 100; 150];
3. V_100_aprox=polyval(V_100_apx, x);
4. plot(x,V_100_aprox);
```

Aproximovaná data jsou uvedena v tabulce. Hodnoty tlaku jsou uvedeny v Bar a hodnoty průtoku jsou uvedeny v m<sup>3</sup>/h.

<b>Aproximovaná data</b>					
Otevření 100%		Otevření 60%		Otevření 50%	
Tlak	Průtok	Tlak	Průtok	Tlak	Průtok
0	0,917481	0	0,820278	0	0,799838
25	1,79	25	1,70	25	1,68
50	2,597317	50	2,484725	50	2,44177
75	3,347413	75	3,160741	75	3,079334
100	4,037629	100	3,732618	100	3,594632
150	5,238417	150	4,563957	150	4,258424
Otevření 40%		Otevření 30%		Otevření 25%	
Tlak	Průtok	Tlak	Průtok	Tlak	Průtok
0	0,796622	0	0,759786	0	0,569552
25	1,54	25	1,38	25	1,18
50	2,210435	50	1,928565	50	1,722314
75	2,80326	75	2,394457	75	2,186656
100	3,320033	100	2,781351	100	2,576306
150	4,125416	150	3,318146	150	3,13153
Otevření 20%		Otevření 15%		Otevření 10%	
Tlak	Průtok	Tlak	Průtok	Tlak	Průtok
0	0,505002	0	0,453757	0	0,348418
25	1,03	25	0,82	25	0,61
50	1,482366	50	1,144205	50	0,838352
75	1,865536	75	1,436568	75	1,042733
100	2,178365	100	1,693691	100	1,220055
150	2,592997	150	2,102214	150	1,493527

Tab. 10 Data aproximovaná polynomem 2.řádu

Na obrázku jsou vykresleny aproximovaná data. Pro porovnání jsou vloženy také naměřené hodnoty. Z tabulek i obrázku lze vidět, že hodnoty neprocházejí bodem (0;0). To nevadí, protože z důvodu poškození čerpadla není možné trvale pracovat s menšími průtoky než 1m<sup>3</sup>/h.



Obr. 34 Data aproximovaná polynomem 2. řádu

Nyní byly hodnoty poskládány tak, aby se v tabulce nacházeli hodnoty otevření pro příslušný průtok při jmenovitém tlaku.

```
1. tlak_25 = [V_100_aprox(2, :) 100; V_60_aprox(2, :) 60; V_50_aprox(2, :) 50; V_40_aprox(2, :) 40; V_30_aprox(2, :) 30; V_25_aprox(2, :) 25; V_20_aprox(2, :) 20; V_15_aprox(2, :) 15; V_10_aprox(2, :) 10; 0 0];
2. tlak_50 = [V_100_aprox(3, :) 100; V_60_aprox(3, :) 60; V_50_aprox(3, :) 50; V_40_aprox(3, :) 40; V_30_aprox(3, :) 30; V_25_aprox(3, :) 25; V_20_aprox(3, :) 20; V_15_aprox(3, :) 15; V_10_aprox(3, :) 10; 0 0];
3. tlak_75 = [V_100_aprox(4, :) 100; V_60_aprox(4, :) 60; V_50_aprox(4, :) 50; V_40_aprox(4, :) 40; V_30_aprox(4, :) 30; V_25_aprox(4, :) 25; V_20_aprox(4, :) 20; V_15_aprox(4, :) 15; V_10_aprox(4, :) 10; 0 0];
4. tlak_100 = [V_100_aprox(5, :) 100; V_60_aprox(5, :) 60; V_50_aprox(5, :) 50; V_40_aprox(5, :) 40; V_30_aprox(5, :) 30; V_25_aprox(5, :) 25; V_20_aprox(5, :) 20; V_15_aprox(5, :) 15; V_10_aprox(5, :) 10; 0 0];
5. tlak_150 = [V_100_aprox(6, :) 100; V_60_aprox(6, :) 60; V_50_aprox(6, :) 50; V_40_aprox(6, :) 40; V_30_aprox(6, :) 30; V_25_aprox(6, :) 25; V_20_aprox(6, :) 20; V_15_aprox(6, :) 15; V_10_aprox(6, :) 10; 0 0];
```

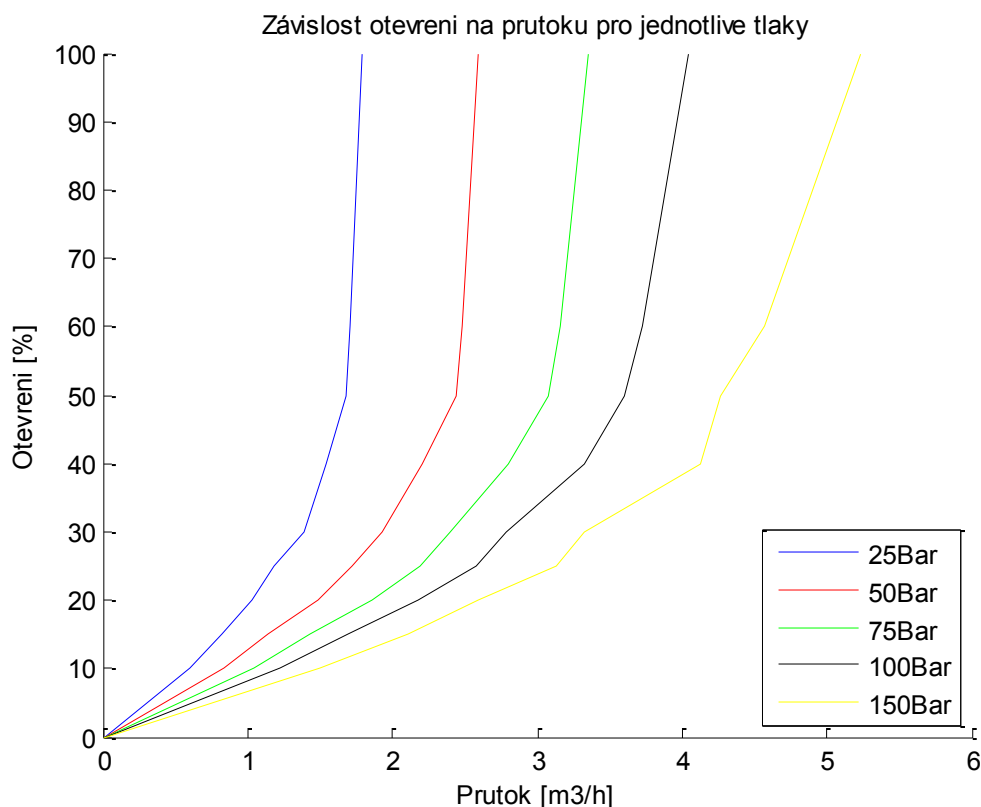


V tabulce jsou uvedeny velikosti průtoku pro jednotlivé otevření při jmenovitém tlaku. Hodnoty tlaku jsou uvedeny v Bar a hodnoty průtoku jsou uvedeny v m<sup>3</sup>/h. Otevření v %

Závislost otevření na průtoku při jmenovitém tlaku					
Tlak 25		Tlak 50		Tlak 75	
Prutok	Otevreni	Prutok	Otevreni	Prutok	Otevreni
1,78734	100	2,597317	100	3,347413	100
1,704571	60	2,484725	60	3,160741	60
1,681938	50	2,44177	50	3,079334	50
1,541555	40	2,210435	40	2,80326	40
1,383674	30	1,928565	30	2,394457	30
1,183279	25	1,722314	25	2,186656	25
1,028855	20	1,482366	20	1,865536	20
0,816601	15	1,144205	15	1,436568	15
0,606914	10	0,838352	10	1,042733	10
0	0	0	0	0	0
Tlak 100		Tlak 150			
Prutok	Otevreni	Prutok	Otevreni		
4,037629	100	5,238417	100		
3,732618	60	4,563957	60		
3,594632	50	4,258424	50		
3,320033	40	4,125416	40		
2,781351	30	3,318146	30		
2,576306	25	3,13153	25		
2,178365	20	2,592997	20		
1,693691	15	2,102214	15		
1,220055	10	1,493527	10		
0	0	0	0		

Tab. 11 Závislost otevření na průtoku při jmenovitém tlaku

Vykreslení aproximovaných dat.



Obr. 35 Závislost otevření na průtoku pro jednotlivé tlaky

Zakřivení křivky pro hodnotu tlaku 150Bar je způsobeno odhadem uvedeným v textu výše. Na laboratorním modelu není za současných podmínek možno dosáhnout takového tlaku při vyšších hodnotách otevření, proto se tahle nepřesnost v provozu neobjeví.

Nebylo možné provést aproximaci  $n$ -tým řádem tak, aby se vstupní data shodovala, proto byly následující hodnoty odečteny z grafu ručně.

		Tlak [Bar]					
		0	25	50	75	100	150
Průtok [M3/h]	0	0	0	0	0	0	0
	0,5	100	8,23	5,96	4,79	4,09	3,34
	1	100	19,32	12,64	9,59	8,19	6,69
	1,5	100	37,36	20,36	15,74	12,95	10,05
	2	100	100	32,53	22,09	18,16	14,16
	2,5	100	100	65,41	32,58	24,04	19,05
	3	100	100	100	47,12	34,05	23,77
	3,5	100	100	100	100	46,55	32,25

Tab. 12 Závislost otevření ventilu na průtoku a vstupním tlaku

V tabulce je uvedena velikost otevření ventilu v % pro žádaný průtok při jmenovitém tlaku. Tabulka také slouží jako vstupní data pro linearizační funkci.

#### 7.4. Návrh linearizační funkce v prostředí STEP7

Jednou z možností pro odstranění nelinearity systému, je zavedení do regulačního řetězce další člen, který tuto nelinearitu potlačí. Jak už bylo napsáno výše, jednou takovou možností je zavedení dalšího regulátoru, který do obvodu vnese další pomocnou regulovanou veličinu, která se mění mnohem rychleji, než regulovaná veličina hlavní. Tento regulátor se bude také muset potýkat s touto nelinearitou, ale její účinky vyruší rychleji a vliv na hlavní regulovanou veličinu sníží.

V případě prvku, jakým je proporcionální ventil, který obsahuje nelinearitu ve své statické charakteristice, se nabízí vytvoření linearizační funkce, která na základě vstupních naměřených dat, vypočte polohu ventilu pro požadovaný průtok, který se tímto stane akční veličinou.

Cílem tedy byl návrh a realizace řízení konstantního průtoku kapaliny při nelineární charakteristice vlastního regulačního ventilu. Proto byla vytvořena linearizační funkce, která tyto požadavky splňuje. Součástí funkce je také implementace adaptivní linearizační křivky ventilu, pro řízení konstantního průtoku kapaliny, při nelineárně proměnlivém vstupním tlaku kapaliny.

V praxi to znamená, že funkce pozná, že do systému vnikla porucha, například změna frekvence otáčení čerpadla a je schopna tuto poruchu vykompenzovat změnou otevření ventilu. Na vstup funkce je tedy nutné přivést informaci o vstupním tlaku před ventilem, bez kterého by implementaci adaptivní křivky nebyla možná. Znalost informace o vstupním tlaku dělá z funkce jakýsi samostatný regulátor.

Principem funkce je použití vzorce pro lineární interpolaci, kdy si funkce v tabulce nalezne interval, ve kterém se hodnota na vstupu nachází, zjistí si krajní hodnoty minima a maxima a poté dopočítá z těchto dvou hodnot a vstupu výstupní hodnotu. Použití právě lineární interpolace bylo vybráno z důvodu snadné matematické implementace. Dosažení požadované přesnosti je zde dáno dostatečným počtem prvků v tabulce. Lineární interpolace se zde používá právě pro doplnění naměřených dat ze vstupní tabulky. Vzorec pro lineární interpolaci je ve funkci ve výpočtu otevření použit celkem 3x.

Naměřená data jsou uložena v datovém bloku, jehož číslo je přivedeno na vstup funkce. Struktura datového bloku je uvedena v tabulce.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	NoOfRow	INT	8
+2.0	NoOfColumn	INT	6
+4.0	XValues	"XValues"	
+36.0	ZValues	"ZValues"	
+60.0	YValues	"YValues"	
=252.0		END_STRUCT	

Obr. 36 Struktura datového bloku, z kterého funkce hledá hodnoty

V prvních dvou řádcích je uveden počet řádků a počet sloupců osy *Y*. Osa *Y* představuje velikosti otevření ventilu. První dva řádky mají datový formát INT, tzn. rozsah 2B. Od adresy 4 jsou uloženy hodnoty jednotlivých os. Nejprve hodnoty osy *X*, která představuje žádaný průtok, poté hodnoty osy *Z*, která představuje vstupní tlak a nakonec hodnoty osy *Y*, které představuje otevření ventilu. Pro přehlednost jsou hodnoty všech os uloženy ve strukturách, které jsou uvedeny v přílohách. Hodnoty os jsou uloženy ve formátu FLOATING POINT, tzn. jejich rozsah je 4B. Celý datový blok má velikost 252B.

Vstupní data mají v tabulce následující strukturu.

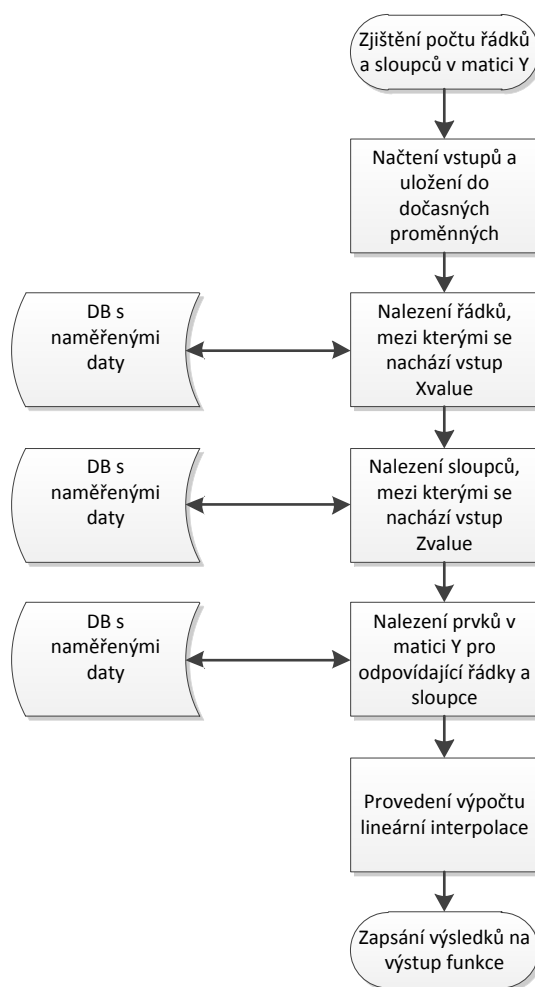
		Tlak [Bar]					
		1	2	3	4	5	6
Průtok [m <sup>3</sup> /h]	index	Zvalues					
	1	Xvalues	Yvalues				
	2						
	3						
	4						
	5						
	6						
	7						
	8						

Tab. 13 Struktura dat uložených v DB

Osu *Yvalues* představuje matice jednotlivých otevření pro žádaný průtok při jmenovitém tlaku na vstupu ventilu. V datovém bloku jsou sloupce této matice poskládány za sebou, tedy za posledním prvkem prvního sloupce následuje první prvek druhého sloupce, za posledním prvkem druhého sloupce následuje první prvek třetího sloupce a tak dále.

Způsob, jakým byla tyto data získána, je popsán v samostatné kapitole této práce.

Algoritmus výpočtu, jaký funkce používá, je znázorněn níže.



Obr. 37 Algoritmus výpočtu výstupní hodnoty funkce

Pro vysvětlení principu funkce uvažujme:

- Žádaný průtok –  $2,1\text{m}^3/\text{h}$
- Vstupní tlak – 113Bar

Nejprve si funkce načte hodnoty ze vstupů, z datového bloku si vyčte počet řádků a sloupců a uloží si je do dočasných proměnných.

```

L    #MatrixDB
T    #DbNumber

OPN  DB [#DbNumber]
L    DBW    0
T    #NoOfRow
L    DBW    2
T    #NoOfColumn

L    #XValue
  
```

```

T      #XVal

L      #ZValue
T      #ZVal

```

Poté se funkce v cyklech pokouší najít hodnoty osy *X*. Nejprve si načte hodnotu z řádku s největším indexem a porovná jí se vstupní hodnotou, tedy s žadáním průtokem.

```

L      #NoOfRow
T      #Index

```

JU\_M

```

_M: L#Index
    L 4
    *I
    SLD      3
    LAR1
    L DBD    [AR1,P#0.0]
    T #AuxVal
    L #XVal
    >R
    JC _M01
    JU _M02

```

Pokud je hodnota žadaného průtoky menší, sníží se *index* řádku o 1 a porovnání se provede znovu, dokud nebude nalezena nemenší hodnota

```

_M01: L      #Index
      L      1
      -I
      T      #Index
      JU      _M

```

Pokud hodnota žadaného průtoky nebude větší než porovnávaná hodnota, funkce zjistí, zda se obě hodnoty rovnají

```

_M02: L      DBD [AR1,P#0.0]
      L      #XVal
      ==R
      JC      _M03
      JU      _M04

```

Pokud se obě hodnoty rovnají, funkce uloží index nalezeného řádku a hodnotu nalezeného řádku.

```
_M03: L      #Index
      T      #RowIndexEq
      L      4
      *I
      SLD    3
      LAR1
      L      DBD [AR1,P#0.0]
      T      #XEq
```

Pokud se hodnoty nerovnají, uloží index současného řádku jako menší hodnotu, a *index* + 1 jako index řádku s vyšší hodnotou. Poté obě hodnoty vyčte z datového bloku a uloží si je do svých dočasných proměnných.

```
_M04: L      #Index
      T      #RowIndexLs
      L      1
      +I
      T      #RowIndexGr

      L      #RowIndexLs
      L      4
      *I
      SLD    3
      LAR1
      L      DBD [AR1,P#0.0]
      T      #XLs

      L      #RowIndexGr
      L      4
      *I
      SLD    3
      LAR1
      L      DBD [AR1,P#0.0]
      T      #XGr
```

Adresa řádku v datovém bloku je vypočítávána pomocí vzorce

$$RowAddress = 4 \cdot RowIndex$$

Pro výše uvedený průtok tedy funkce nalezne řádky:

		Tlak [Bar]					
		0	25	50	75	100	150
Průtok [m <sup>3</sup> /h]	0	0	0	0	0	0	0
	0,5	100	8,23	5,96	4,79	4,09	3,34
	1	100	19,32	12,64	9,59	8,19	6,69
	1,5	100	37,36	20,36	15,74	12,95	10,05
	2	100	100	32,53	22,09	18,16	14,16
	2,5	100	100	65,41	32,58	24,04	19,05
	3	100	100	100	47,12	34,05	23,77
	3,5	100	100	100	100	46,55	32,25

Tab. 14 Nalezené řádky pro uvedený příklad

Pomocí stejného algoritmu funkce nalezne také odpovídající sloupce. Jediný rozdíl je ten, že jsou adresy v datovém bloku vypočítávány pomocí vzorce:

$$ColumnAddress = (NoOfRow + ColumnIndex) \cdot 4$$

Pro hodnotu 113Bar funkce nalezne sloupce:

		Tlak [Bar]					
		0	25	50	75	100	150
Průtok [m <sup>3</sup> /h]	0	0	0	0	0	0	0
	0,5	100	8,23	5,96	4,79	4,09	3,34
	1	100	19,32	12,64	9,59	8,19	6,69
	1,5	100	37,36	20,36	15,74	12,95	10,05
	2	100	100	32,53	22,09	18,16	14,16
	2,5	100	100	65,41	32,58	24,04	19,05
	3	100	100	100	47,12	34,05	23,77
	3,5	100	100	100	100	46,55	32,25

Tab. 15 Nalezené sloupce pro uvedený příklad

V dalším kroku funkce pomocí nalezených indexů řádků a sloupců vyčte z datového bloku hodnoty osy Y. Celkem vyčte 4 hodnoty

```

L      #ColIndexLs
L      1
-I
L      #NoOfRow
*I
L      #RowIndexLs
+I
L      #IndexOffset
+I
L      4

```



\*I  
SLD 3  
LAR1  
L DBD[AR1,P#0.0]  
T#y11

Adresy jednotlivých hodnot jsou vypočítávány pomocí vzorce:

$$YAddress = \{[NoOfRow \cdot (ColumnIndex - 1) + RowIndex] + [NoOfRow + NoOfColumn]\} \cdot 4$$

Pro vstupní hodnoty 2,1m<sup>3</sup>/h a 113Bar jsou tedy nalezeny hodnoty:

		Tlak [Bar]					
		0	25	50	75	100	150
Průtok [M3/h]	0	0	0	0	0	0	0
	0,5	100	8,23	5,96	4,79	4,09	3,34
	1	100	19,32	12,64	9,59	8,19	6,69
	1,5	100	37,36	20,36	15,74	12,95	10,05
	2	100	100	32,53	22,09	18,16	14,16
	2,5	100	100	65,41	32,58	24,04	19,05
	3	100	100	100	47,12	34,05	23,77
	3,5	100	100	100	100	46,55	32,25

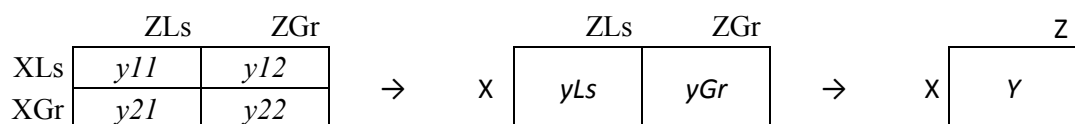
Tab. 16 Nalezené hodnoty otevření pro uvedený příklad

Všechny nalezené hodnoty jsou v proměnných uloženy jako:

		Tlak [Bar]					
						ZLs	ZGr
Průtok [m3/h]	XLs XGr						
						y11	y12
						y21	y22

Tab. 17 Význam proměnných použitých ve výpočtu

Poté je provedena lineární interpolace. Celkem je provedena 3x. Nejprve pro hodnotu tlaku 100Bar a poté pro hodnotu 150Bar. Tím je získána interpolovaná hodnota pro požadovaný průtok pro oba krajní tlaky. Následně je provedena interpolace těchto dvou hodnot pro tlak 113Bar.



Tab. 18 Postup výpočtu výstupní hodnoty  $Y$

Interpolace je počítána obecně pomocí vzorce:

$$Output = \frac{(Input - Xmin) \cdot \Delta Y}{\Delta X} + Ymin$$

Ukázka výpočtu pro finální hodnotu je uvedena níže

```

L      #YGr
L      #YLs
-R
T      #DeltaY

L      #ZGr
L      #ZLs
-R
T      #DeltaX

L      #ZVal
L      #ZLs
-R
L      #DeltaY
*R
L      #DeltaX
/R
L      #YLs
+R
T      #Result

```

Pokud byl předtím nalezen řádek nebo sloupec, který byl roven, neprovádí se interpolace 3x, ale pouze jednou. Výsledek interpolace je zapsán do dočasné proměnné *Result*.

Funkce také obsahuje části kódu, ve které kontroluje hodnoty na vstupech. Účelem kontroly je zjistit, zda vstupní hodnoty leží v mezích os  $X$  a  $Z$ . To má zabránit nechtěnému chybnému zadání. To by ve výsledku mohlo způsobit chybu, která by přivedením na výstup funkce mohla zasáhnout do regulačního procesu a tím snížit její kvalitu, nebo za použití v průmyslových aplikacích způsobit další škody.

Kontrola vstupů se provádí úplně na začátku. Nejprve zkontroluje, zda hodnota na vstupu *XValue* není menší než nejmenší hodnota osy  $X$ , pokud ano, zapíše do proměnné, do které se kopíruje hodnota vstupu *XValue* právě nejmenší hodnotu osy  $X$ . Stejný postup provede také s hodnotou ze vstupu *ZValue*, na který v případě, že hodnota ze vstupu *ZValue* bude menší než nejmenší hodnota osy  $Z$ , zapíše do proměnné, do které se zapisuje hodnota vstupu *ZValue* nejmenší hodnotu osy  $Z$ .

Poté je provedena kontrola maxim. Funkce porovná hodnotu vstupu *XValue* s největší hodnotou osy *X*, a pokud je vyšší, zapíše do proměnné, do které se zapisuje hodnota vstupu *XValue* nejvyšší hodnotu osy *X*. Poté porovná hodnotu vstupu *ZValue* a pokud je vyšší než největší hodnota osy *Z*, zapíše do proměnné, do které se zapisuje hodnota vstupu *ZValue* nejvyšší hodnotu osy *Z*.

Poté, co funkce nalezne indexy a hodnoty všech řádků a sloupců, zapíše si také příznaky, podle kterých se rozhoduje, zda použije výpočet pouze pro jeden řádek, pokud se hodnota vstupu *XValue* shoduje s některou z hodnot osy *X*, nebo pouze pro jeden sloupec, pokud se hodnota vstupu *ZValue* shoduje s některou z hodnot osy *Z*. V tomto případě hodnota interpoluje pouze ve dvou osách, *X* a *Y* popřípadě *Z* a *Y*. Pokud nenalezne shodu ani v ose *X*, ani v ose *Z*, funkce interpoluje ve všech třech osách. K takovému rozhodování slouží právě příznaky. Pomocí nich je také možno kontrolovat správnost výpočtu při tvorbě řídicího programu.

Úplně nakonec jsou na výstup funkce zapsány hodnoty z dočasných proměnných. Jedná se o všechny hodnoty os *X*, *Y* a *Z*, které funkce používá ke svému výpočtu. Pokud je na vstupu hodnota, která je rovna některé z hodnot os *X* a *Z*, je do obou výstupů *XLsValue* a *XGrValue*, popřípadě *ZLsValue* a *ZGrValue* zapsána stejná hodnota, která je rovna vstupu *XValue* popřípadě *XGrValue*.

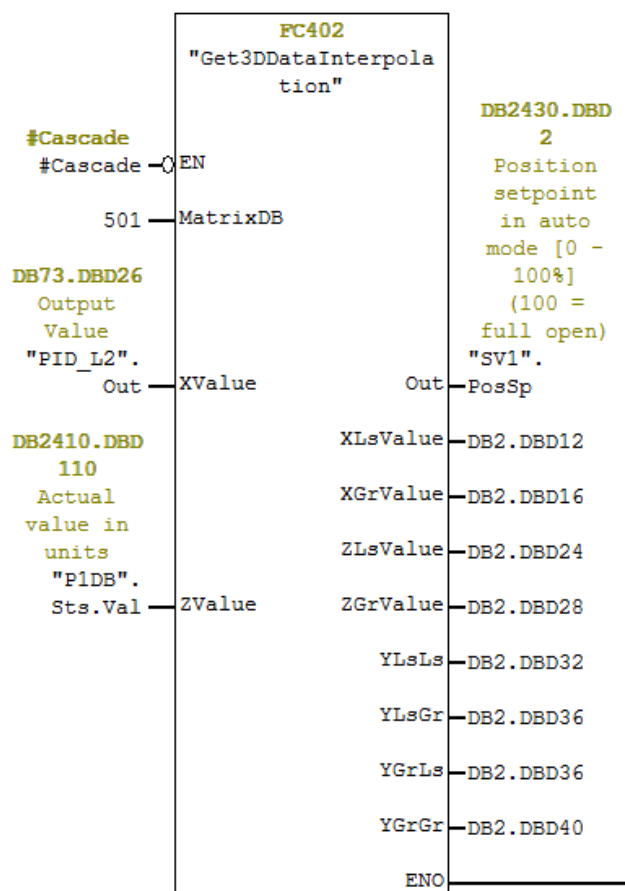
Schématické znázornění významu výstupů je uvedeno v následující tabulce

	ZLsValue	ZGrValue
XLsValue	<i>YLsLs</i>	<i>YLsGr</i>
XGrValue	<i>YGrLs</i>	<i>YGrGr</i>

Tab. 19 Schématické znázornění výstupů linearizační funkce

Úplně nakonec, je na výstup *Out* zapsána hodnota z proměnné *Result*, ve které je uložen výsledek výpočtu.

Funkce vypadá v FBD následovně:



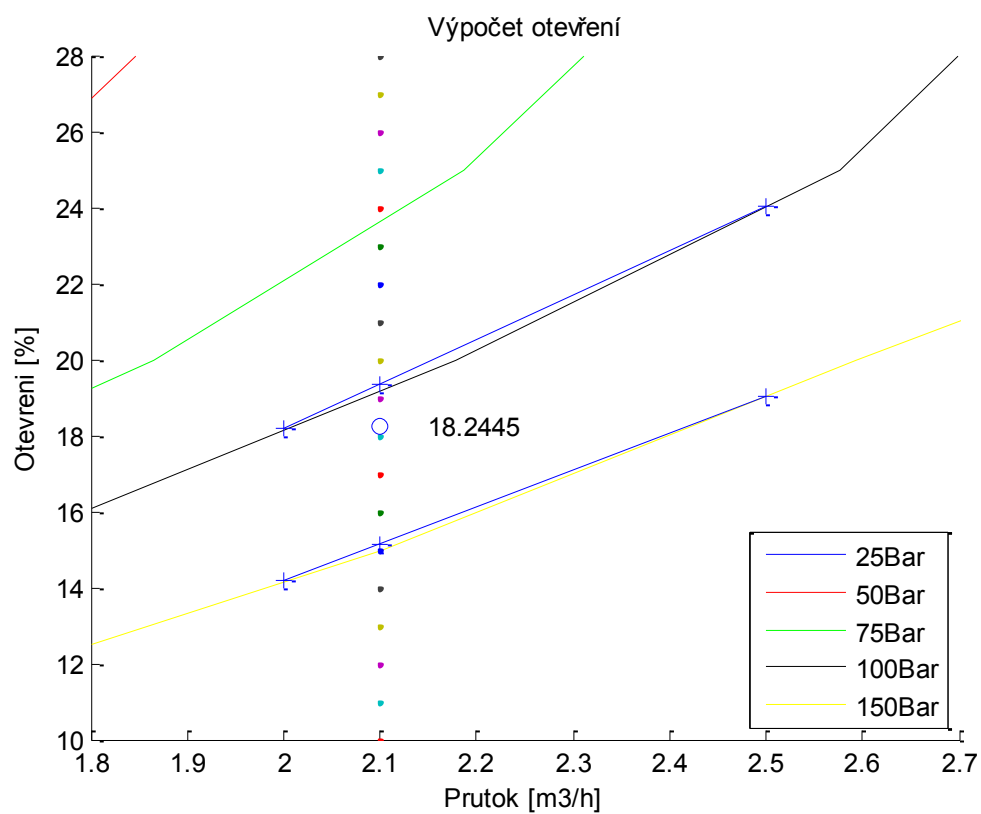
Obr. 38 Vzhled vytvořené linearizační funkce v FBD

Tabulka vstupů a výstupů linearizační funkce:

Vstup	Formát	Význam
<i>Matrix DB</i>	<i>Int</i>	Číslo datového bloku s naměřenými daty
<i>Xvalue</i>	<i>Float</i>	Hodnota osy X (žádaný průtok)
<i>Zvalue</i>	<i>Float</i>	Hodnota osy Z (tlak na vstupu ventlu)
Výstup	Formát	Význam
<i>Out</i>	<i>Float</i>	Interpolovaná hodnota (otevření ventilu)
<i>XLsValue</i>	<i>Float</i>	Nalezená nižší hodnota osy X
<i>XGrValue</i>	<i>Float</i>	Nalezená vyšší hodnota osy X
<i>ZLsValue</i>	<i>Float</i>	Nalezená nižší hodnota osy Z
<i>ZGrValue</i>	<i>Float</i>	Nalezená vyšší hodnota osy Z
<i>YLSls</i>	<i>Float</i>	Hodnota osy Y pro XLsValue a ZLsValue
<i>YLSGr</i>	<i>Float</i>	Hodnota osy Y pro XLsValue a ZGrValue
<i>YGrLs</i>	<i>Float</i>	Hodnota osy Y pro XGrValue a ZLsValue
<i>yGrGr</i>	<i>Float</i>	Hodnota osy Y pro XGrValue a ZGrValue

Tab. 20 Tabulka vstupů a výstupů funkce

# Grafické znázornění výpočtu



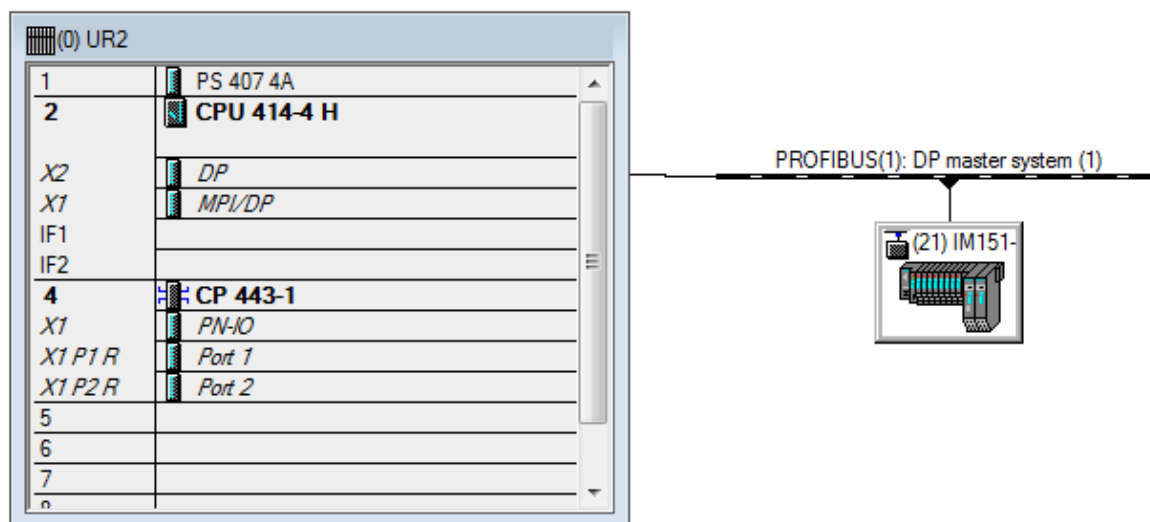
Obr. 39 Grafické znázornění výpočtu výstupu funkce

## 8. Hardwarová konfigurace

Vstupy a výstupy školícího zařízení „Vodní svět“ jsou přivedeny do decentrální periferie SIMATIC ET 200S. Jednotka komunikuje s řídicím systémem po sběrnici PROFIBUS. Adresa jednotky na sběrnici je nastavena na 21.

Jako řídicí systém bylo používáno CPU 414 – 4H. CPU je napájeno zdrojem PS 407 který je schopen dávat proud až 4A. CPU dokáže komunikovat po sběrnici PROFIBUS. Na racku je také umístěn ethernetový komunikační procesor CP 443-1, který sloužil k programování a komunikaci s vizualizací. IP adresa komunikačního procesoru byla nastavena na 10.10.10.100.

K vizualizaci sloužil osobní počítač, na kterém bylo nainstalováno prostředí WINCC Flexible.



Obr. 40 Hardwarová konfigurace

### 8.1. Decentrální periferie SIMATIC ET 200S

SIMATIC ET 200S je vzdálených vstupů/výstupů komunikující po sběrnici PROFIBUS. Nachází se v rozvaděči vedle zařízení „Vodní svět“ a jsou na ni přivedeny všechny vstupy a výstupy ze zařízení. Celkem obsahuje 9 jednotek vstupů nebo výstupů a dvě napájecí jednotky ve slotech 2 a 6.

Níže jsou uvedeny všechny jednotky, které se nacházejí na periférii ET 200S. V tabulkách jsou uvedeny jednotlivé vstupy a výstupy zařízení „Vodní svět“, které jsou na danou jednotku připojeny.

#### A2

- PM-E DC24..48V
- Zdroj 24V – 48V, 10A

#### A3

- 4DI DC24V ST
- 4 digitální vstupy
- Rozsah adres – I0.0 až I0.3

Adresa vstupu	Připojené zařízení
I0.0	Relé výstup průtokoměru F1
I0.1	Výstup 1 průtokoměru F1
I0.2	Výstup 2 průtokoměru F1
I0.3	Průtokový ohřívač PO1 zapnuto

Tab. 21 Adresy jednotky A3

#### A4

- 4DI DC24V ST
- 4 digitální vstupy
- Rozsah adres – I1.0 až I1.3

Adresa vstupu	Připojené zařízení
I1.0	Průtokový ohřívač PO1 napájení OK
I1.1	oběhové čerpadlo M1 chod
I1.2	oběhové čerpadlo M1 napájení OK
I1.3	rezerva

Tab. 22 Adresy jednotky A4

#### A5

- 2DO DC24V/0.5A HF
- 2 digitální výstupy
- Rozsah adres – Q0.0 až Q0.1

Adresa vstupu	Připojené zařízení
Q0.0	Průtokový ohřívač PO1 zapnout
Q0.1	Oběhové čerpadlo M1 chod

Tab. 23 Adresy jednotky A5

#### A6

- PM-E DC24..48V

#### A7

- 2AI I 4WIRE HS
- 2 analogové vstupy
- 15bitů + znaménko pro rozsah +-20mA
- Rozsah - +-20mA, 0 až 20mA, 4 až 20mA
- Rozsah adres – PIW512 až PIW515

Adresa vstupu	Připojené zařízení
PIW512	Tlakový snímač P1
PIW514	Tlakový snímač P2

Tab. 24 Adresy jednotky A7

## A8

- 2AI I 4WIRE HS
- Rozsah adres – PIW516 až PIW518

Adresa vstupu	Připojené zařízení
PIW516	Hladinový snímač L1
PIW518	Hladinový snímač L2

Tab. 25 Adresy jednotky A8

## A9

- 2AI I 4WIRE HS
- Rozsah adres – PIW520 až PIW522

Adresa vstupu	Připojené zařízení
PIW520	Průtokoměr F1
PIW522	Regulační ventil RV1

Tab. 26 Adresy jednotky A9

## A10

- 2AI I 4WIRE HS
- Rozsah adres – PIW524 až PIW526

Adresa vstupu	Připojené zařízení
PIW524	Teplota v nádrži T3
PIW526	Rezerva

Tab. 27 Adresy jednotky A10

## A11

- 2AI RTD ST
- 2 analogové vstupy pro odporové teploměry nebo odporové měření
- Rozsah adres – PIW528 až PIW530

Adresa vstupu	Připojené zařízení
PIW528	Teplota před ohříváčem T1
PIW530	Teplota za ohříváčem T2

Tab. 28 Adresy jednotky A11



## **A12**

- 2AO I ST
- 2 analogové výstupy
- Rozsah adres – PQW512 až PQW514

<b>Adresa vstupu</b>	<b>Připojené zařízení</b>
PQW512	Žádaná hodnota regulačního ventilu RV1
PQW514	Rezerva

Tab. 29 Adresy jednotky A12

## 9. Řídicí program v PLC

Program byl navržen v prostředí SIMATIC STEP7. Jedná se o software pro programování programovatelných automatů z rodiny SIMATIC S7. Z důvodu velkého rozšíření těchto automatů se jedná o nejrozšířenější software pro programování programovatelných automatů a v konkurenci s jinými programovacími nástroji se stal standardem.

Samotný řídicí program byl vytvořen v programovacím jazyku FBD (diagram funkčních bloků). Jedná se o grafický jazyk pro tvorbu programů pro programovatelné automaty, kde se program vytváří pomocí blokového schématu. Výhodou jazyku je možnost vytvořit a opakovaně používat uživatelské funkční bloky. FBD je jeden z programovacích jazyků, které definuje mezinárodní norma IEC 61131-3.

Program obsahuje celkem čtyři regulátory. Tři pro regulaci výšky hladiny v nádrži a jeden pro regulaci průtoku přítoku do nádrže. Dva z regulátorů regulující výšku hladiny ji regulují pomocí otevření ventilu a jejich akční zásah je tedy procentuální otevření proporcionálního ventilu. Třetí regulátor regulující výšku hladiny pracuje ve dvou režimech. V prvním režimu vstupuje akční veličina, kterou je průtok přítoku do nádrže do linearizační funkce, jejíž výstupem je linearizované procentuální otevření ventilu. Ve druhém režimu je aktivována kaskádní regulace, pomocí regulátoru průtoku.

Řídicí program byl vytvořen pomocí zvyklostí a standardů firmy Ingeteam a.s. s použitím jejich vlastních funkčních bloků pro obsluhu jednotlivých prvků zařízení „Vodní Svět.“

Program obsahuje několik organizačních bloků (OB) obsluhující různé chyby. V OB100, bloku který je volán po restartu CPU, je nastaven bit StrUp, který v programu definuje první cyklus PLC. Tento bit je v logické 1 pouze v průběhu prvního cyklu PLC, poté je nastaven na log.0.

Hlavní cyklický blok OB1 je rozdělen na 3 funkce, ProgramBegin, UserProgram a ProgramEnd. ProgramBegin obsahuje nastavení bitů log0 a log1, jsou to bity, které jsou po celý běh programu nastaveny na příslušnou logickou úroveň. Dále je zde nastavováno několik bitů s různou frekvencí. Na konci OB1 je resetován bit StrUp.

### 9.1. UserProgram

Funkce UserProgram obsahuje blok FB300 „VodniSvet“ v kterém je implementován hlavní řídicí program. Na začátku bloku jsou zkopírovány příkazy z vizualizace a vymazána struktura s chybami. Poté následuje měření analogových hodnot pro monitorování stavu zařízení. Celkem je provedeno 8 analogových měření: měření tlaku před a za ventilem, průtok zařízením, výška hladiny v přečerpávací i záchytné nádrži a pro úplnost teplota za průtokovým ohříváčem a teploty v obou nádržích.

Dále v programu následuje rozhodování, zda bude výška hladiny regulována průtokem, nebo otevřením ventilu. V případě regulace průtokem, je možnost výběru mezi regulací průtoku dalším regulátorem (kaskádní regulace) nebo pomocí linearizační funkce. V případě regulace pomocí otevření ventilu je možnost přepínání mezi dvěma regulátory, u kterých je možnost nastavení různých konstant.

Jednotlivé regulátory jsou vloženy v následujících čtyřech částech kódu (Network). Pokud je vybráno nastavení otevření ventilu pomocí linearizační funkce, je aktivní tento blok.

Pro nastavení otevření je vložen blok obsluhující proporcionální ventil. Protože samotný ventil má poloviční zdvih než servopohon, je zde také implementován přepočet, který zajistí, aby při požadovaném otevření 0 až 100% bylo skutečné otevření v těchto mezích.

Čerpadlo je spouštěno blokem obsluhující střídavé motory bez měniče.

Dále jsou do stavové struktury zapsány proměnné, které jsou odesílány do vizualizace. Odesílání je prováděno na konci programu.

Níže jsou uvedeny struktury příkazů a stavů sloužící pro komunikaci s vizualizací

Příkaz	Typ	Význam
Ack	BOOL	Kvitace poruchy
RqStr	BOOL	Start
RqStp	BOOL	Stop
RqCas	BOOL	Kaskádní regulace
RqInt	BOOL	Interpolace
RqVlv1	BOOL	Regulátor otevření ventilu 1
RqVlv2	BOOL	Regulátor otevření ventilu 2
RqFlwCtrl	BOOL	Regulace průtokem
RqVlvCtrl	BOOL	Regulace otevření ventilu

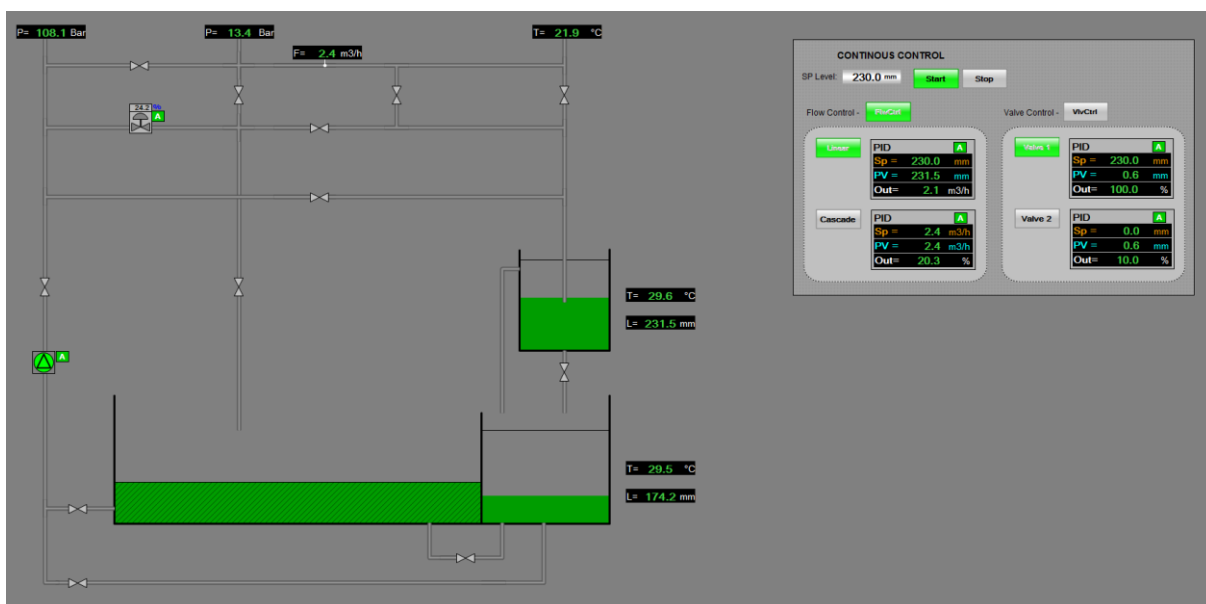
Tab. 30 Struktura příkazů

Příkaz	Typ	Význam
Flt	BOOL	Chyba
FltNotAck	BOOL	Nepotvrzení chyby
Rdy	BOOL	Připraveno
RdyInAuto	BOOL	Připraveno - všechny bloky v auto módě
Process	BOOL	Spuštěno
Cascade	BOOL	Kaskádní regulace
Stpd	BOOL	Zastaveno
Lin	BOOL	Zapnuta linearizace
Vlv1	BOOL	Regulátor otevření ventilu 1
Vlv2	BOOL	Regulátor otevření ventilu 2
FlwCtrl	BOOL	Regulace průtokem
VlvCtrl	BOOL	Regulace ventilem

Tab. 31 Struktura stavů

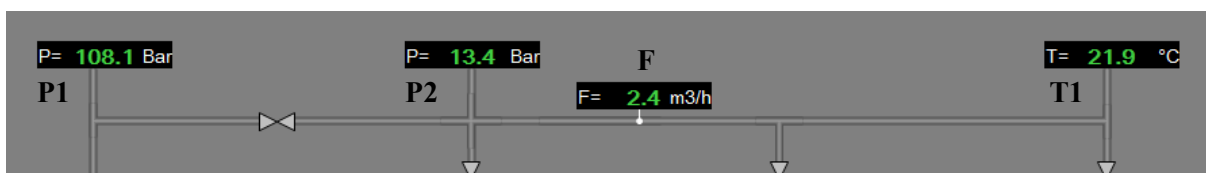
## 10. Vizualizace

Pro jednoduché ovládání a sledování zařízení „Vodní svět“, byla vytvořena jednoduchá vizualizace v prostředí WINCC Flexible.

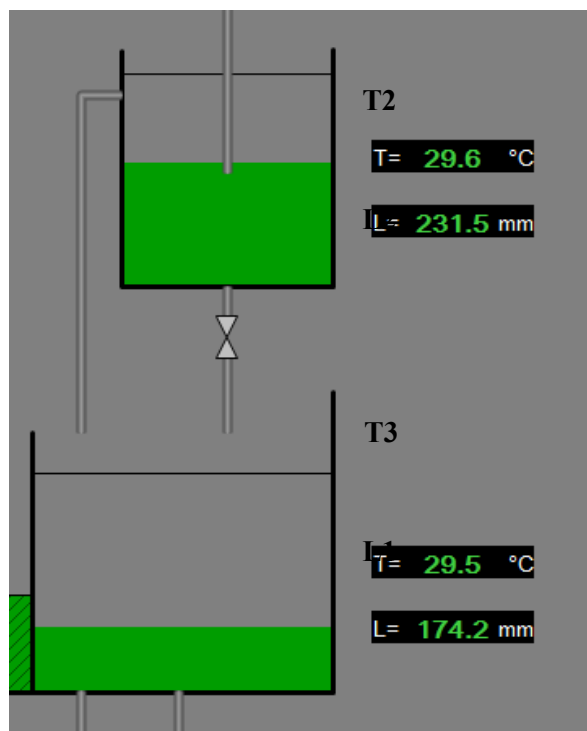


Obr. 41 Vizualizace systému

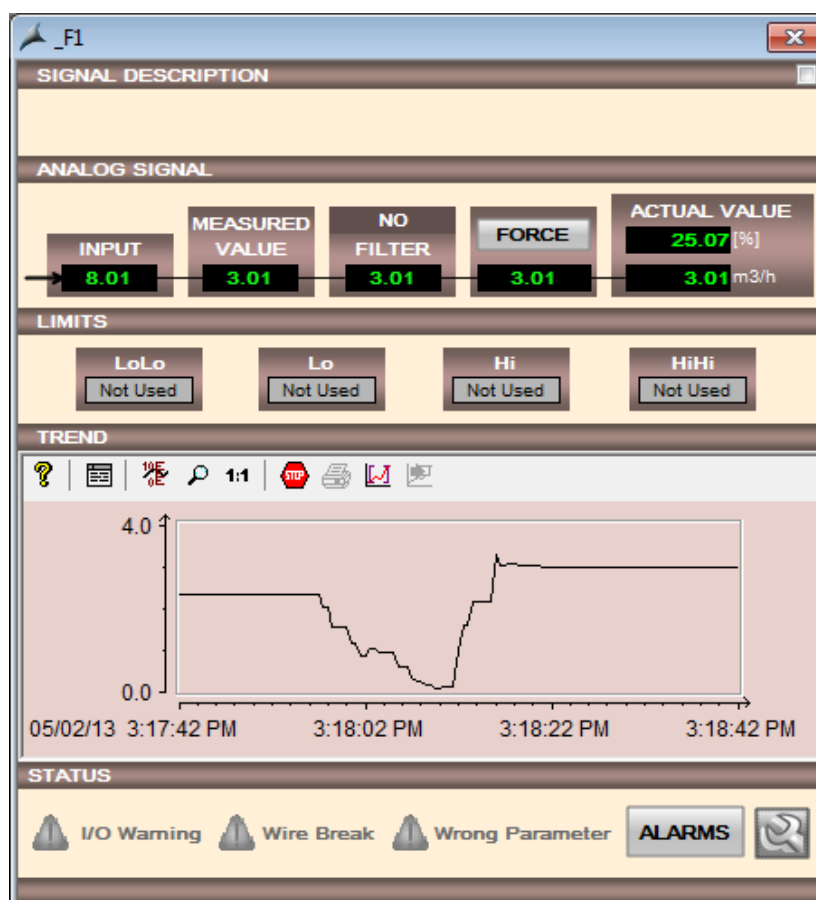
Levá část obsahuje zjednodušený model zařízení. Jsou zde vidět všechny analogové veličiny, které jsou použity pro monitorování stavu.



Obr. 42 Analogové měření veličin v potrubí



Obr. 43 Analogové měření stavu obou nádrží

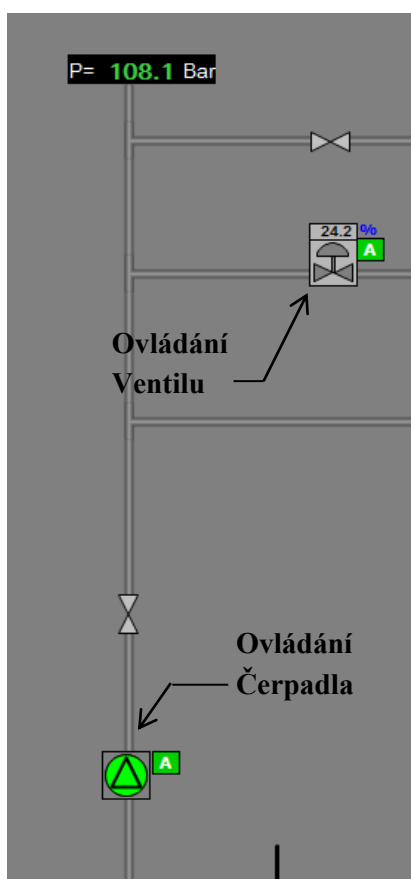


Obr. 44 Okno pro analogové měření

Veličina	Význam
$P1$	Tlak před proporcionálním ventilem
$P2$	Tlak za proporcionálním ventilem
$F$	Průtok potrubím
$L1$	Výška hladiny záchytné jímky
$L2$	Výška hladiny přečerpávací nádrže
$T1$	Teplota vody za průtokoměrem
$T2$	Teplota vody v přečerpávací nádrži
$T3$	Teplota vody v záchytné jímce

Tab. 32 Měřené veličiny

V obvodu jsou vloženy dva bloky pro ruční ovládání čerpadla a proporcionálního ventilu. Oba bloky slouží pro manuální ovládání zařízení.



Obr. 45 Bloky pro ruční ovládání zařízení



Obr. 46 Bloky pro ovládání čerpadla a proporcionálního ventilu

Pravá část vizualizace slouží k ovládání regulace. Žádaná výška je zadávána změnou hodnoty textového pole *SP Level*. Tlačítka *Start* a *Stop* slouží k spuštění a zastavení celého procesu. Podmínkou spuštění je, aby všechny bloky byli v automatickém módu. Jinak je tlačítko *Start* neaktivní. Zelená barva tlačítka znázorňuje, v jakém stavu se zařízení nachází (zelené tlačítko *Start* = zařízení spuštěno). K dispozici je výběr pro regulaci průtokem, nebo regulaci otevřením ventilu. K tomu slouží tlačítka *FlwCtrl* pro regulaci průtokem a *VlvCtrl* pro regulaci otevřením ventilu. Který způsob regulace je vybrán, reprezentuje opět zelená barva tlačítka.

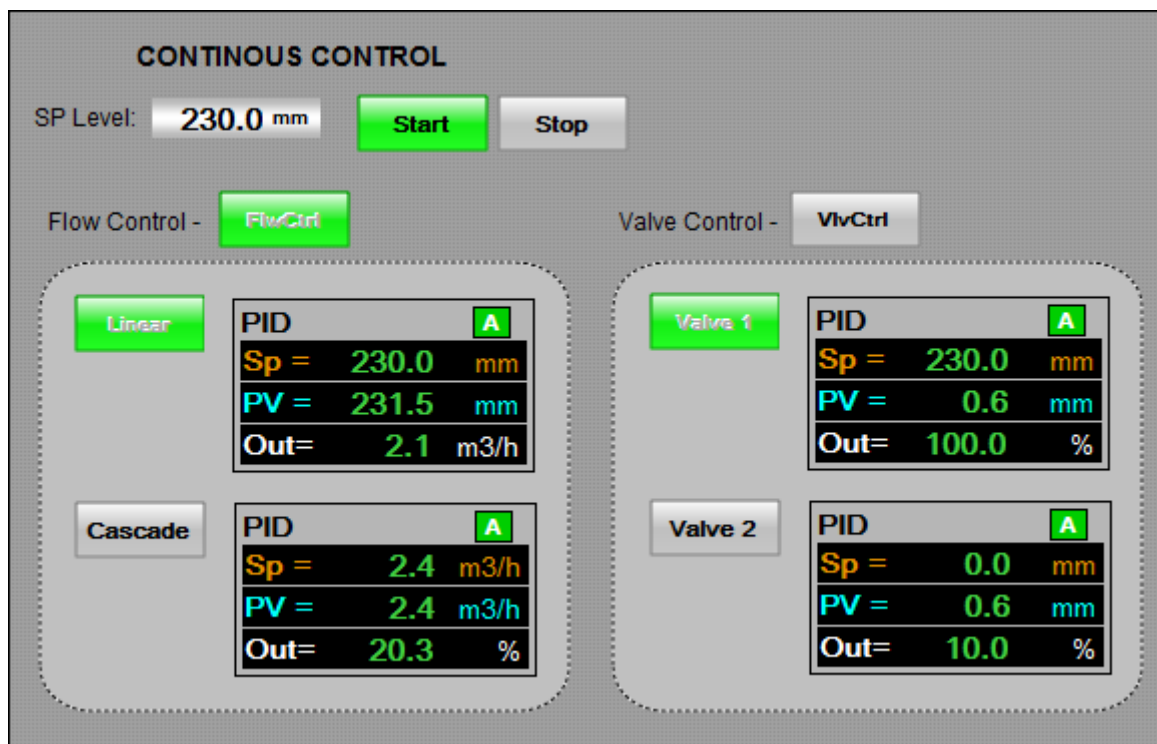
### 10.1. Regulace průtokem

Regulace průtokem funguje ve dvou režimech. Prvním je použití linearizační funkce. Pro výběr použití linearizační funkce musí být aktivováno tlačítko *Linear*. Výstupem regulátoru je poté žádaný průtok, který vstupuje do linearizační funkce, jejíž výstupem je otevření ventilu.

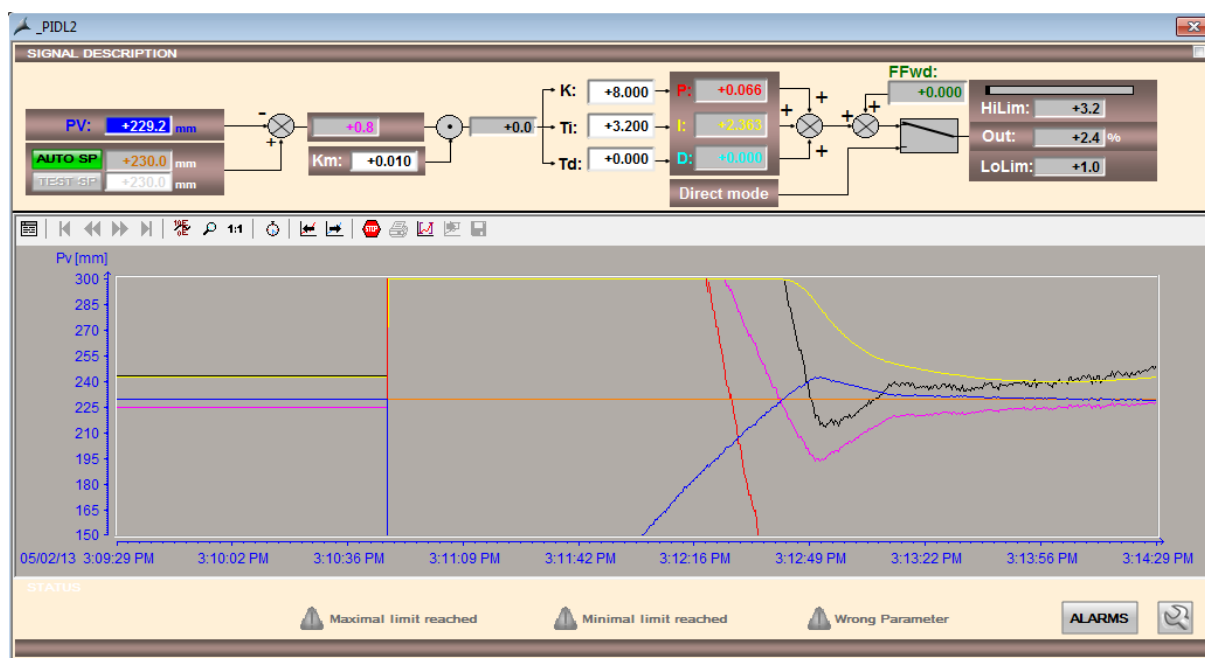
Ve druhém režimu, který je spuštěn tlačítkem *Cascade*, je výstup regulátoru připojen na vstup druhého regulátoru. Výstupem toho regulátoru je otevření ventilu odpovídající žádanému průtoku.

### 10.2. Regulace otevření ventilu

Výstupem obou těchto regulátorů je otevření ventilu. Oba dva regulátory mají výstup připojen přímo na žádanou hodnotu otevření bloku obsluhující proporcionální ventil. U každého z regulátorů mohou být konstanty nastaveny nezávisle. Průtok není v tomto případě vůbec sledován.



Obr. 47 Ovládání regulace



Obr. 48 Blok regulátoru

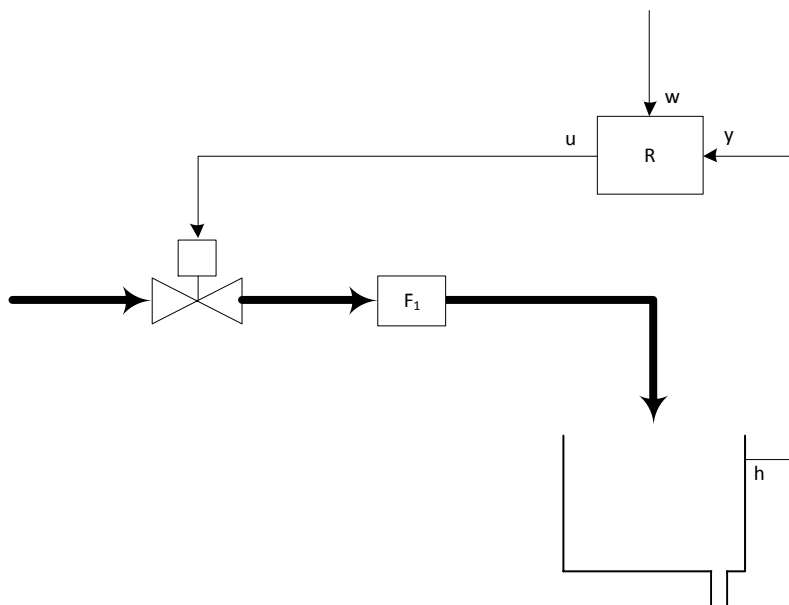


## 11. Zhodnocení výsledků

Pro měření byla nastavena výška hladiny 230mm, poté byl nastaven odtok z nádrže  $2,4\text{m}^3/\text{h}$ . Otevření ventilu při tomto průtoku je asi 20%. V tomto pracovním bodě byly nastaveny konstanty regulátoru a změřena přechodová charakteristika. Poté byl odtok nastaven na  $2\text{m}^3/\text{h}$  a změřena odezva při zvýšení výšky hladiny o 10mm. Otevření ventilu při průtoku  $2\text{m}^3/\text{h}$  je asi 15%. To stejné bylo provedeno při odtoku  $2,6\text{m}^3/\text{h}$ . Otevření ventilu při průtoku  $2,6\text{m}^3/\text{h}$  je asi 27%. Cílem, bylo sledovat čas, kdy při těchto malých změnách v rozdílných pracovních bodech bylo dosaženo žádané hodnoty. Podmínkou bylo, aby ani při v jednom pracovním bodě při změně o 10mm nedosáhl výstup regulátoru saturace.

### 11.1. Regulace pomocí otevření ventilu

První způsob regulace výšky hladiny v přečerpávací nádrži je regulace pomocí otevření ventilu. Výstup regulátoru je přímo připojen na vstup bloku obsluhující proporcionální ventil.



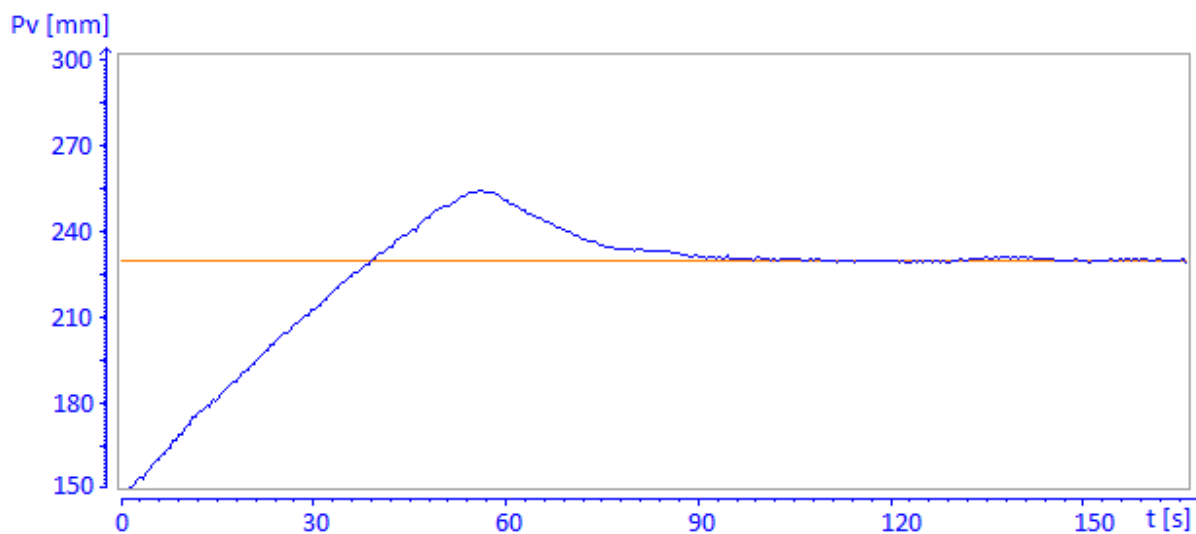
Obr. 49 Blokové schéma regulace pomocí otevření ventilu

Signál	Rozsah	Jednotka	Význam
w	0 - 300	mm	požadovaná výška hladiny
y	0 - 300	mm	skutečná výška hladiny
u	5 - 100	%	otevření ventilu

Tab. 33 Signály regulace pomocí otevření ventilu

Výška hladiny je nastavována v rozmezí 0 – 300mm, poté dojde k odtoku vody do pomocné trubky zabráňující přetečení přes hrany nádrže. Otevření ventilu je nastavováno v rozmezí 5 – 100%. Při menším otevření může při delším provozu dojít k poškození čerpadla.

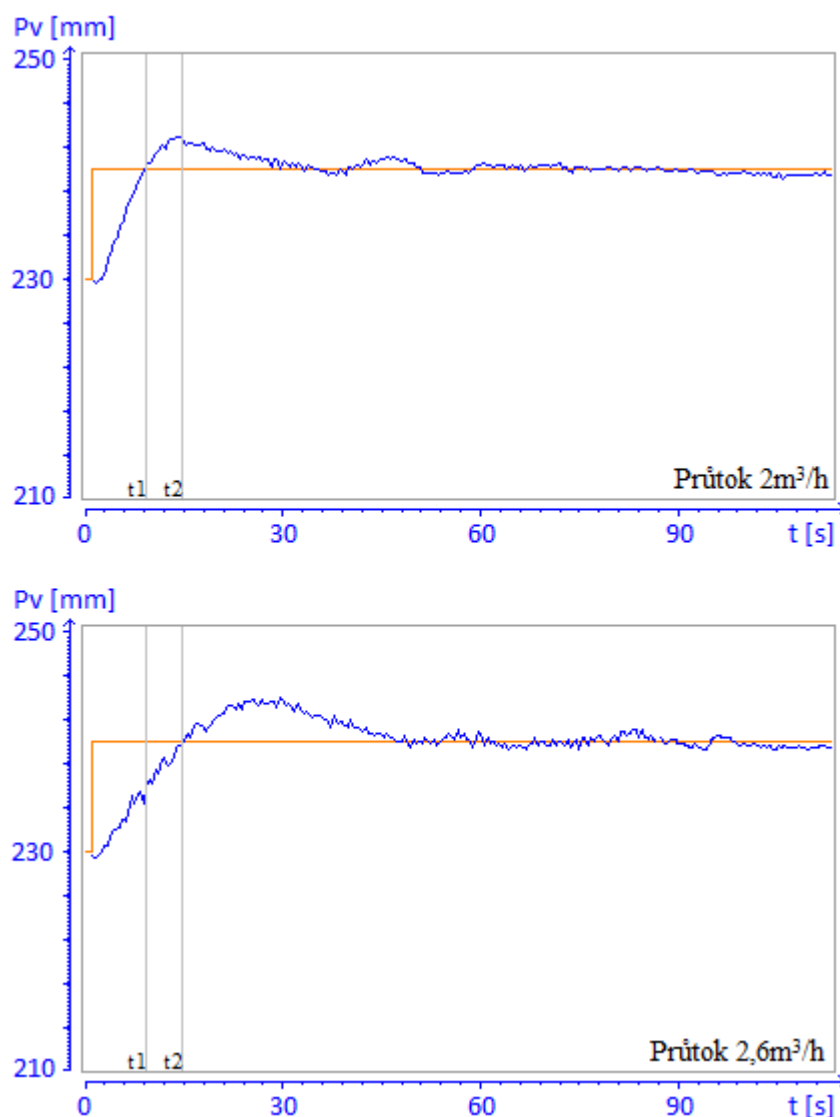
### Odezva regulace otevření ventilu



Obr. 50 Odezva při regulaci pomocí otevření ventilu

Okamžitě po startu regulace přechází akční zásah do saturace. Na obrázku je patrný překmit (25mm), který je způsoben nelinearitou proporcionálního ventilu. Je to způsobeno časem, kdy ventil přechází přes část otevření, kdy se průtok mění pouze minimálně.

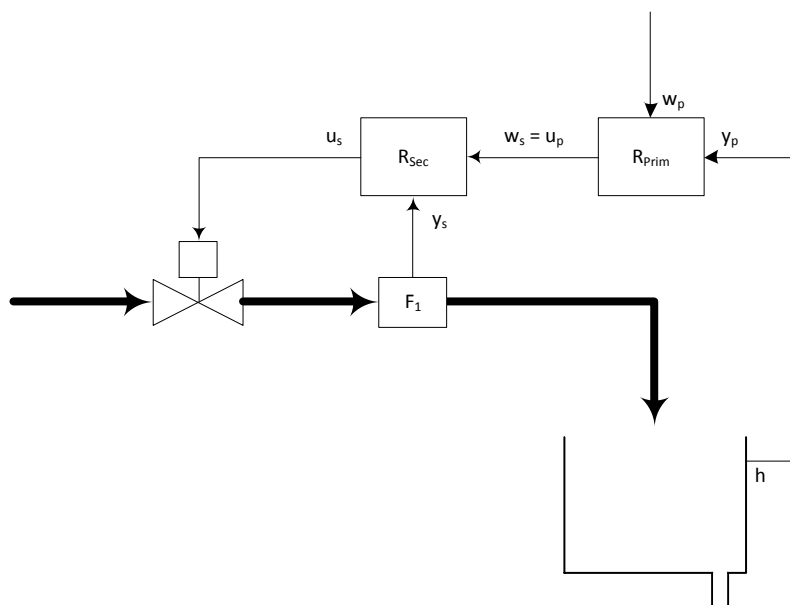
Na odezvách o 10mm v různých pracovních bodech je vidět, že při průtoku  $2,6\text{m}^3/\text{h}$  bylo dosaženo žádané veličiny o cca 6s později, než při průtoku  $2\text{m}^3/\text{h}$ . To je způsobeno nelinearitou ventilu. Regulátor si v obou případech žádá stejnou procentuální změnu otevření ventilu, která ale není úměrná skutečnému přítoku do nádrže. Při průtoku  $2\text{m}^3/\text{h}$ , kdy je ventil otevřen na 15% znamená změna otevření o 5% změnu průtoku o  $0,4\text{m}^3/\text{h}$ . Při průtoku  $2,6\text{m}^3/\text{h}$  kdy je ventil otevřen na 27% ale změna otevření o 5% je rovna změně průtoku o 0,2%. To má za následek celkové zpomalení regulace.



Obr. 51 Odezva na změnu o 10mm při různých průtocích při regulaci otevření ventilu

## 11.2. Kaskádní regulace

Další způsob regulace výšky hladiny v přečerpávací nádrži je kaskádní regulace. Do obvodu je zavedena nová pomocná regulovaná veličina – průtok. Do obvodu byl přidán sekundární regulátor, který pomocí otevření ventilu reguluje množství vody přitékající do nádrže, tedy průtok. Výstup sekundárního regulátoru je připojen na vstup bloku ovládající proporcionální ventil. Akční veličinou je tedy otevření ventilu. Sledovanou veličinou je hodnota průtoku, kterou poskytuje průtokoměr. Výstupem primárního regulátoru je průtok, který je zároveň žádanou veličinou sekundárního regulátoru.

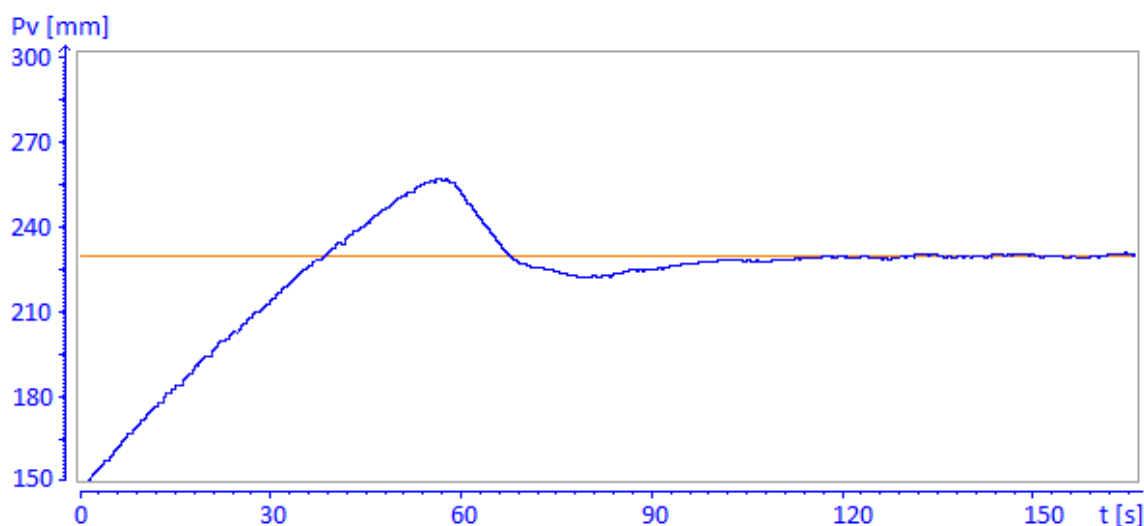


Obr. 52 Blokové schéma kaskádní regulace

Signál	Rozsah	Jednotka	Význam
$w_p$	0 - 300	mm	požadovaná výška hladiny
$y_p$	0 - 300	mm	skutečná výška hladiny
$u_p = w_s$	1 - 3,2	$\text{m}^3/\text{h}$	žádaný průtok
$y_s$	0 - 3,2	$\text{m}^3/\text{h}$	skutečný průtok
$u_s$	5 - 100	%	žádané otevření ventilu

Tab. 34 Signály kaskádní regulace

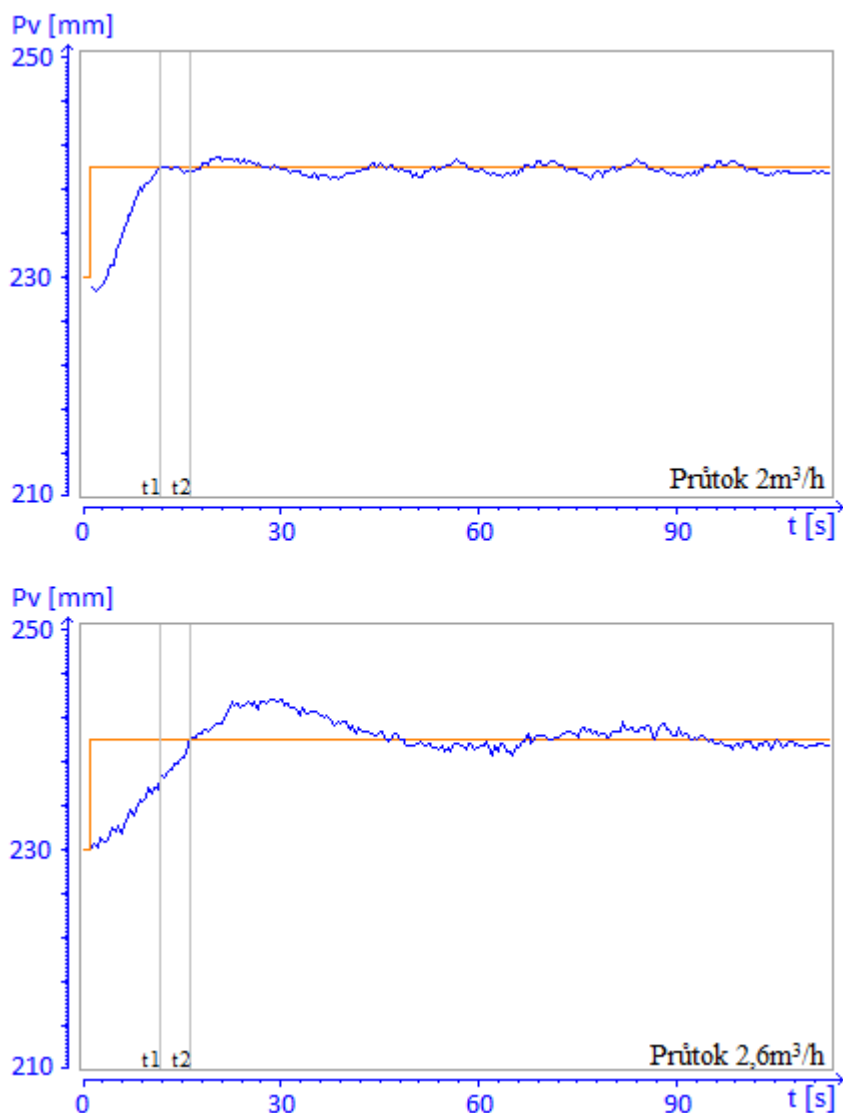
### Odezva kaskádní regulace



Obr. 53 Přejchodová charakteristika kaskádní regulace

Překmit je zde podobný jako u regulace pomocí otevření ventilu. Na obrázku je vidět také druhý překmit, který je ovšem podstatně menší než ten první. Doba ustálení se oproti prvnímu případu

prodloužila o cca 10s. Druhý překmit a prodloužení regulačního děje je dáno dopravním zpožděním celého systému, které je znásobenou použitím druhého regulátoru. Také v tomto případě přechází výstup obou regulátorů ihned po zahájení děje do saturace.

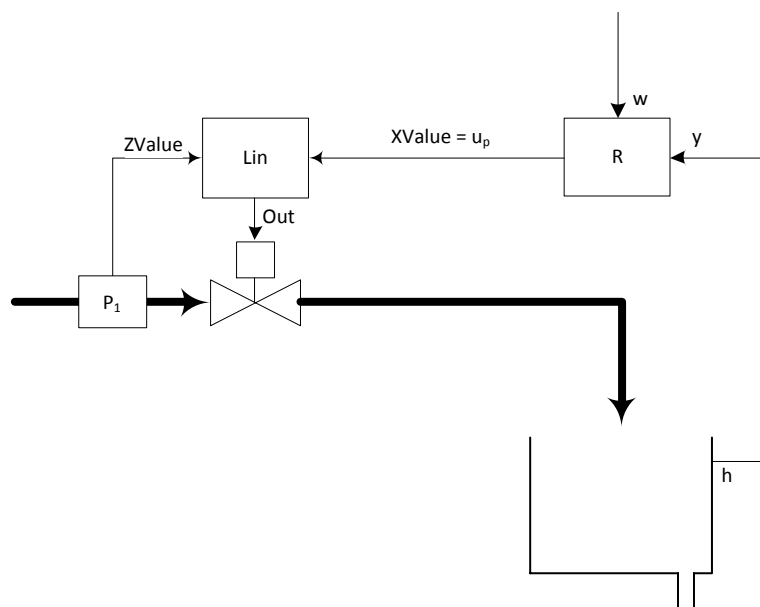


Obr. 54 Odezva na změnu o 10mm při různých průtocích při kaskádní regulaci

Časový rozdíl dosažení žádané hodnoty mezi různými pracovními body se při kaskádní regulaci zmenšil pouze nepatrně (cca 5s). Z toho lze usuzovat, že potlačení nelinearity proporcionálního ventilu kaskádní regulace v tomto případě neřeší. Jednou z podmínek kaskádní regulace je, aby rychlost sekundární regulace byla minimálně 5x rychlejší, než rychlost primární regulace. To v tomto případě nebylo možno dodržet, protože elektricky poháněný ventil nemá dostatečnou rychlost přenastavení své polohy. V součtu se znásobeným dopravním zpožděním je toto řešení nevhodné a zbytečně složité. Navíc při změně pracovního bodu soustava mírně kmitá. Kmitání je způsobeno ventilem, který nedokáže reagovat na malé změny otevření.

### 11.3. Použití linearizační funkce

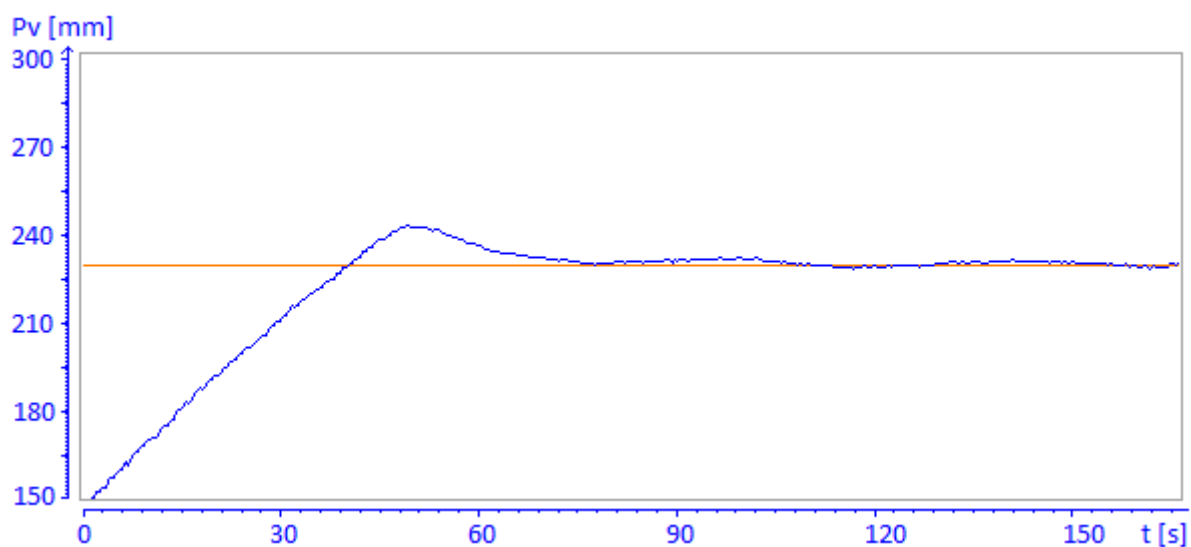
Posledním způsobem je použití vytvořené linearizační funkce. Výstup regulátoru je zapojen do vstupu linearizační funkce jako žádaná hodnota průtoku. Informace o tlaku před ventilem je získána z tlakoměru zde umístěným. Výstupem linearizační funkce je otevření, odpovídající žádané hodnotě průtoku při jmenovitém tlaku před ventilem.



Obr. 55 Blokové schéma regulace s použitím linearizační funkce

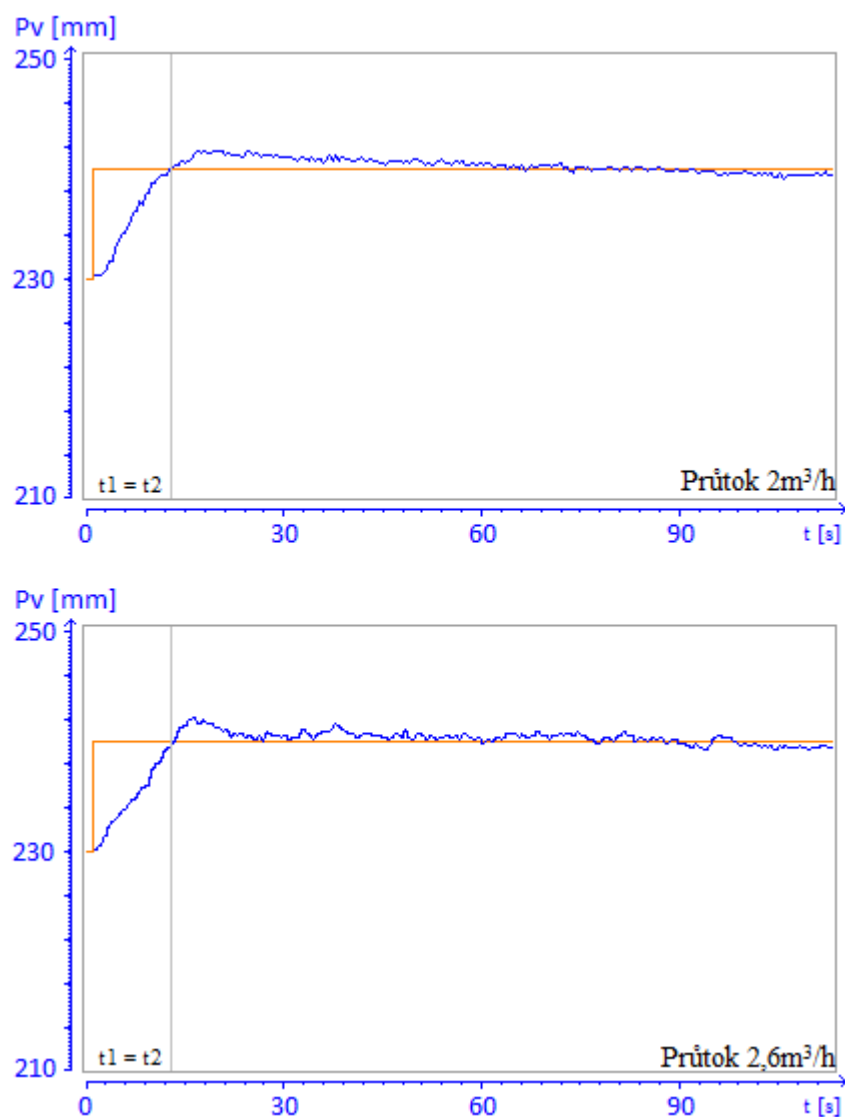
Signál	Rozsah	Jednotka	Význam
w	0 - 300	mm	požadovaná výška hladiny
y	0 - 300	mm	skutečná výška hladiny
$u_p = XValue$	1 - 3,2	m <sup>3</sup> /h	žádaný průtok
ZValue	70 - 160	Bar	tlak před ventilem
Out	5 - 100	%	otevření ventilu

Tab. 35 Signály regulace s použitím linearizační funkce



Obr. 56 Přejchodová charakteristika regulace s použitím linearizační funkce

Ze všech použitých způsobů má použití linearizační funkce nejmenší překmit (cca 12mm). Doba ustálení je také nejmenší, přibližně o 12s kratší než při regulaci otevření ventilu. Také zde přechází akční veličina regulátoru ihned do saturace, ale její návrat je podstatně kratší než v obou předchozích případech. To má za následek nejmenší překmit ze všech použitých způsobů.



Obr. 57 Odezva na změnu o 10mm při různých průtocích při použití linearizační funkce

Z obrázku je patrné, že časy, kterých je dosaženo žádané hodnoty v obou pracovních bodech, jsou stejné. Regulátor si v obou případech žádá stejnou změnu průtoku, která je zvolenou interpolací naměřených hodnot dosažena. To je důkaz, že vytvořená linearizační funkce skutečně odstraňuje nelinearitu proporcionálního ventilu. Stejně jako u kaskádní regulace je zde patrná neschopnost ventilu reagovat na malé změny otevíření. To se projevilo mírnou odlišností charakteristik.



## **Závěr**

V diplomové práci se zabývám návrhem a implementací řídicích algoritmů s PID regulátory. K realizaci využívám reálný laboratorní model firmy Ingeteam a.s. „Vodní svět,“ na kterém byl vytvořen program pro regulaci výšky hladiny. K regulaci průtoku zařízením a tedy i přítoku do nádrže je používán proporcionální ventil se servopohonem.

Problém proporcionálního ventilu je ten, že jeho charakteristika je nelineární. V praxi to znamená, že stejná změna otevření ventilu, neznamená stejnou změnu průtoku v různých pracovních bodech. To se projeví pomalejší reakcí celého procesu při změně požadované výšky hladiny při větších hodnotách průtoku a otevření. Cílem bylo tedy navrhnout mechanismus, který tuto nelinearitu odstraní.

V práci byla realizována dvě řešení. Kaskádní regulace a linearizace. V teoretické práci jsou proto shrnuty požadavky a možnosti pro implementaci PID algoritmů v praxi. Zvláštní pozornost je věnována právě kaskádní regulaci a linearizaci. V části o linearizaci je nastíněno několik možných řešení problému, v části o kaskádní regulaci jsou shrnuty výhody, nevýhody a podmínky. Dále je v teoretické části vložena také popis školícího zařízení „Vodní svět“, popis jeho prvků a odvození matematicko-fyzikálního modelu nádrže s nuceným odtokem. Z modelu nádrže vyplývá, že se jedná o nelineární soustavu. Důvodem nelinearity je, že se odtok z nádrže mění s druhou odmocninou výšky hladiny. Proto byla soustava linearizována v okolí pracovního pomoci nástrojů programu Matlab. Výsledkem byla soustava prvního řádu. Proto k regulaci postačil jednoduchý PI regulátor.

V praktické části byla obě řešení implementována do řídicího programu. Jako řídicí systém byl použit SIMATIC S7 400 s decentralní periferií ET 200S komunikující s CPU pomocí sběrnice PROFIBUS.

Nejprve byla realizována kaskádní regulace. Do obvodu byla přidána nová veličina - průtok a k primárnímu regulátoru, který reguluje výšku hladiny v nádrži, byl přidán sekundární regulátor, který reguluje objemový průtok vody přicházející od nádrže. Akční zásah primárního regulátoru byl přiveden na vstup žádané hodnoty sekundárního regulátoru, který měl zajistit pomocí údaje z instalovaného průtokoměru požadovanou velikost průtoku. Tím měla být zajištěna stejná odezva systému ve všech pracovních bodech.

Druhým řešením bylo vytvoření linearizační funkce. Požadavkem bylo, aby funkce dokázala reagovat na změnu tlaku na vstupu proporcionálního ventilu. Funkce byla vytvořena v jazyce STL v prostředí STEP 7. Výsledkem je hodnota otevření, která odpovídá požadovanému průtoku v závislosti na tlaku před proporcionálním ventilem. Hodnota otevření je vypočtena pomocí lineární interpolace hodnot, které byly pro účely funkce naměřeny a upraveny. Postup měření a úprava naměřených dat je popsána v textu. Aby byla funkce univerzální, jsou naměřená data uložena v DB a funkce z nich pouze vyčítá hodnoty potřebné k výpočtu. Linearizační funkce byla poté implementována do regulačního řetězce.

V závěru práce jsou porovnány všechny způsoby použité regulace. Celkem jsou uvedeny tři charakteristiky pro každý způsob. Měření probíhalo tak, že byly všechny regulátory nastaveny při stejném průtoku a změřeny jejich odezvy při napouštění nádrže z nulové hodnoty. Poté byly změřeny odezvy na malou změnu výšky hladiny při menším průtoku a to stejné při průtoku větším. Pozorovány

byly časy, kdy výška hladiny dosáhne žádané hodnoty. Podmínkou bylo, aby při těchto změnách nepřecházel regulátor do saturace.

Při přímé regulaci, kdy akční veličinou regulátoru je otevření ventilu je vidět zpomalení odezvy na zásah regulátoru. Při menším průtoku dosáhne výška hladiny požadovaného stavu rychleji než při vyšším otevření. To je způsobeno již zmiňovanou nelinearitou ventilu, kdy stejná změna otevření znamená menší změnu průtoku při vyšších hodnotách otevření. Na odezvě z nulové hodnoty je vidět velký překmit, který je daný tím, že ventil dlouho přechází pásmo, kdy se průtok mění pouze nepatrně.

Při kaskádní regulaci je vidět, že sekundární regulátor problém nevyřešil. Časový rozdíl dosažení požadované hodnoty mezi jednotlivými pracovními body se zmenšil pouze nepatrně. Při větším průtoku obvod navíc mírně kmitá. Překmit na odezvě z nulové hodnoty se sice mírně snížil, ale objevil se také druhý překmit.

Podmínkou kaskádní regulace je, aby rychlost sekundární regulace byla alespoň 5x vyšší, než rychlost primární regulace. To ale elektricky poháněný ventil nedokáže zajistit, protože je jeho doba přenastavení velká. Proto je toto řešení pro daný typ pohonu nevhodné a uplatnění by našlo při použití jiného rychlejšího typu pohonu, např. pneumatického.

Problémy vyřešila až vytvořená linearizační funkce. Překmit při odezvě z nulové hodnoty se snížil na polovinu a při odezvách na malou změnu při různých hodnotách průtoku je vidět, že čas dosažení požadované hodnoty je stejný. Z toho je vidět správnost výsledků linearizační funkce.

## Seznam literatury

- [1] OŽANA, Štěpán. *Navrhování a realizace regulátorů*. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2012, 2 DVD-ROM. ISBN 978-80-248-2605-9.
- [2] BENNETT, Stuart. *A history of control engineering: 1800-1930*. 1979, 214 s. ISBN 978-0-86341-047-5.
- [3] ŠULC, Bohumil. *Teorie a praxe návrhu regulačních obvodů*. 1. vyd. Praha: Vydavatelství ČVUT, 2004, 333 s. ISBN 80-010-3007-5.
- [4] KLÁN, Petr. *Metody zlepšení PI regulace*. Automa. 2001, č. 12. Dostupné z: (5/2013) <<http://www.odbornecasopisy.cz/>>.
- [5] Technická dokumentace k D2411. Dostupné z (5/2013) <<http://www.jsp.cz/>>.
- [6] Technická dokumentace k LMK351. Dostupné (5/2013) z <<http://www.jsp.cz/>>.
- [7] Technická dokumentace k FLONET. Dostupné (5/2013) z <<http://www.elis.cz/>>.
- [8] Technická dokumentace k SQX62. Dostupné (5/2013) z <<http://www.siemens.cz/>>.
- [9] *Aproximace funkcí*. Plzeň, 2001. Učební text centra aplikované matematiky. Západočeská univerzita v Plzni. Dostupné (5/2013) z <<http://www.cam.zcu.cz/>>.

## **Seznam příloh**

### **Příloha I: Výpis programu linearizační funkce**

- a) Vstupy funkce
- b) Výstupy funkce
- c) Pomocné proměnné
- d) Program

### **Příloha II: Struktury použité v DB s daty pro linearizační funkci**

- a) Struktura XValues
- b) Struktura ZValues
- c) Struktura YValues

### **Příloha III: Obsah přiloženého CD**

## Příloha I: Výpis programu linearizační funkce

### a) Vstupy funkce

Name	Data Type	Comment
MatrixDB	Int	Number of DB which contain matrix with values
XValue	Real	Input value of X axis
ZValue	Real	Input value of Z axis

### b) Výstupy funkce

Name	Data Type	Comment
Out	Real	Calculated output value
XLsValue	Real	Finded less value of X axis
XGrValue	Real	Finded great value of X axis
ZLsValue	Real	Finded less value of Z axis
ZGrValue	Real	Finded great value of Z axis
YLSls	Real	Finded Y value of less value of X axis and less value of Z axis
YLSgr	Real	Finded Y value of less value of X axis and great value of Z axis
YGrLs	Real	Finded Y value of great value of X axis and less value of Z axis
YGrGr	Real	Finded Y value of greatvalue of X axis and great value of Z axis

### c) Pomocné proměnné

Name	Data Type	Address	Comment
XGrFound	Bool	0.0	Great value of X axis was found
XLsFound	Bool	0.1	Less value of X axis was found
XeqFound	Bool	0.2	Equal value of X axis was found
ZLsFound	Bool	0.3	Less value of Z axis was found
ZGrFound	Bool	0.4	Great value of Z axis was found
ZeqFound	Bool	0.5	Equal value of Z axis was found
DbNumber	Int	2.0	Number of DB containing input matrix
NoOfRow	Int	4.0	Number of rows
NoOfColumn	Int	6.0	Number of columns
AuxIndex	Int	8.0	Auxiliary index
Index	Int	10.0	Index of row or column in matrix
IndexOffset	Int	12.0	Index Offset
RowIndexEq	Int	14.0	Index of row, if equal value was found
RowIndexLs	Int	16.0	Index of row for less value
RowIndexGr	Int	18.0	Index of row for great value
ColIndexEq	Int	20.0	Index of column, if equal value was found
ColIndexLs	Int	22.0	Index of column for less value
ColIndexGr	Int	24.0	Index of column for great value
XVal	Real	26.0	Value of Xvalue input
ZVal	Real	30.0	Value of Zvalue input
AuxVal	Real	34.0	Auxiliary value
y11	Real	38.0	Found Y value of less value of X axis and less value of Z axis
y12	Real	42.0	Found Y value of less value of X axis and great value of Z axis
y21	Real	46.0	Found Y value of great value of X axis and less value of Z axis
y22	Real	50.0	Found Y value of great value of X axis and great value of Z axis
yC1	Real	54.0	Value of less column if equal row was found
yC2	Real	58.0	Value of great column if equal row was found
yR1	Real	62.0	Value of less row if equal column was found
yR2	Real	66.0	Value of great row if equal column was found
Result	Real	70.0	Result of calculation
DeltaY	Real	74.0	Ymax - Ymin
DeltaX	Real	78.0	Xmax - Xmin
XGr	Real	82.0	Great value of X axis
XLs	Real	86.0	Less value of X axis
XEq	Real	90.0	Equal value of X axis
ZGr	Real	94.0	Great value of Z axis
ZLs	Real	98.0	Less value of Z axis
ZEq	Real	102.0	Equal value of Z axis
YGr	Real	106.0	Great Y value after first interpolation
YLs	Real	110.0	Great Y value after first interpolation

#### d) Program

**Network 1**: Title:

Get matrix information and check inputs

```
L      #MatrixDB
T      #DbNumber

OPN    DB [#DbNumber]
L      DBW      0
T      #NoOfRow
L      DBW      2
T      #NoOfColumn

L      #XValue
T      #XVal

L      #ZValue
T      #ZVal

C0:    L      #NoOfRow
        L      4
        *I
        SLD    3
        LAR1
        L      DBD [AR1,P#0.0]
        L      #XVal
        <R
        JC     C01

        L      #NoOfRow
        L      #NoOfColumn
        +I
        L      4
        *I
        SLD    3
        LAR1
        L      DBD [AR1,P#0.0]
        L      #ZVal
        <R
        JC     C02

        JU     C1

C01:   L      DBD [AR1,P#0.0]
        T      #XVal
        JU     C0

C02:   L      DBD [AR1,P#0.0]
        T      #ZVal
        JU     C0
```

```

C01:  L    DBD [AR1,P#0.0]
      T    #XVal
      JU    C0

C02:  L    DBD [AR1,P#0.0]
      T    #ZVal
      JU    C0

C1:   L    1
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      L    #XVal
      >R
      JC    C11

      L    #NoOfRow
      L    1
      +I
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      L    #ZVal
      >R
      JC    C12

      JU    N1E

C11:  L    DBD [AR1,P#0.0]
      T    #XVal
      JU    C1

C12:  L    DBD [AR1,P#0.0]
      T    #ZVal
      JU    C1

N1E:  NOP  0

```



## Network 2 : Finding rows

Finding indexes of rows and saving less index into RowIndexLs, great index into RowIndexGr and equal index into RowIndexEq if equal values is find.  
Also save values of X axis into Xls, XGr and XEq.

```

      L      #NoOfRow
      T      #Index

      JU      _M

_M:    L      #Index
      L      4
      *I
      SLD    3
      LAR1
      L      DBD [AR1,P#0.0]
      T      #AuxVal
      L      #XVal
      >R
      JC      _M01
      JU      _M02

_M01:  L      #Index
      L      1
      -I
      T      #Index
      JU      _M

_M02:  L      DBD [AR1,P#0.0]
      L      #XVal
      ==R
      JC      _M03
      JU      _M04

_M03:  L      #Index
      T      #RowIndexEq
      L      4
      *I
      SLD    3
      LAR1
      L      DBD [AR1,P#0.0]
      T      #XEq

      L      -1
      T      #RowIndexLs
      T      #RowIndexGr
      JU      _N2E
```

```

_M04: L    #Index
      T    #RowIndexLs
      L    1
      +I
      T    #RowIndexGr

      L    #RowIndexLs
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #XLs

      L    #RowIndexGr
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #XGr

      L    -1
      T    #RowIndexEq

_N2E: NOP  0

```

### Network 3: Finding of columns

Finding indexes of columns and saving less index into ColIndexLs, great index into ColIndexGr and equal index into ColIndexEq if equal values is find. Also save values of Z axis into Zls, ZGr and ZEq.

Adress in DB structure is calculated by pattern:  
 $(\text{NoOfRow} + \text{Index}) * 4 = \text{Column Address}$

```

L    #NoOfColumn
T    #Index
JU   _M1

```

```

_M1:  L    #NoOfRow
      L    #Index
      +I
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #AuxVal
      L    #ZVal
      >R
      JC   _M11

      JU   _M12

_M11: L    #Index
      L    1
      -I
      T    #Index
      JU   _M1

_M12: L    DBD [AR1,P#0.0]
      L    #ZVal
      ==R
      JC   _M13
      JU   _M14

_M13: L    #Index
      T    #ColIndexEq
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #ZEq

      L    -1
      T    #ColIndexLs
      T    #ColIndexGr

      JU   _N3E

```

```

_M14: L    #Index
      T    #ColIndexLs
      L    1
      +I
      T    #ColIndexGr

      L    #NoOfRow
      L    #ColIndexLs
      +I
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #ZLs

      L    #NoOfRow
      L    #ColIndexGr
      +I
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #ZGr

      L    -1
      T    #ColIndexEq

_N3E: NOP  0

```

Network 4 : Signs for decision logic

Set signs for decision logic

```

R    #XLsFound
R    #XGrFound
R    #XeqFound
R    #ZLsFound
R    #ZGrFound
R    #ZeqFound

```

```

L      #RowIndexLs
L      -1
>I
S      #XLsFound

L      #RowIndexGr
L      -1
>I
S      #XGrFound

L      #RowIndexEq
L      -1
>I
S      #XeqFound

L      #ColIndexLs
L      -1
>I
S      #ZLsFound

L      #ColIndexGr
L      -1
>I
S      #ZGrFound

L      #ColIndexEq
L      -1
>I
S      #ZeqFound

```

#### Network 5: Get Y values

Get Y values from matrix and saves into y11 for less row and less column, y12 for less row and great column, y21 for great row and less column and y22 for great row and great column.

```

..    ..    ZLs  ZGr  ..    ..    ..    ZLs  ZGr  ..    ..    ..    ZEq  ..    ..
..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..
XLs   ..    y11  y12  ..    or  XEq  ..    yC1  YC2  ..    or  XLs  ..    YR1  ..    ..
XGr   ..    y21  y22  ..    ..    ..    ..    ..    ..    ..    XGr  ..    YR2  ..    ..
..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..    ..

```

Adresses of Y values is find by pattern:

```

{[NumberOfRow * (IndexOfColl - 1) + IndexOfRow] + [NumberRow +
NumberCollumn]} * 4 = Adress

```

```

L    #NoOfRow
L    #NoOfColumn
+I
T    #IndexOffset

A    #XeqFound
O    #ZeqFound
JC   EQ0

```

```
//For Y[1 1]
```

```

L    #ColIndexLs
L    1
-I
L    #NoOfRow
*I
L    #RowIndexLs
+I
L    #IndexOffset
+I
L    4
*I
SLD  3
LAR1
L    DBD [AR1,P#0.0]
T    #y11

```

```
//For Y[1 2]
```

```

L    #ColIndexGr
L    1
-I
L    #NoOfRow
*I
L    #RowIndexLs
+I
L    #IndexOffset
+I
L    4
*I
SLD  3
LAR1
L    DBD [AR1,P#0.0]
T    #y12

```

```
//For Y[2 1]
```

```
L      #ColIndexLs
L      1
-I
L      #NoOfRow
*I
L      #RowIndexGr
+I
L      #IndexOffset
+I
L      4
*I
SLD    3
LAR1
L      DBD [AR1,P#0.0]
T      #y21
```

```
//For Y[2 2]
```

```
L      #ColIndexGr
L      1
-I
L      #NoOfRow
*I
L      #RowIndexGr
+I
L      #IndexOffset
+I
L      4
*I
SLD    3
LAR1
L      DBD [AR1,P#0.0]
T      #y22

JU     N5E
```

//If equal vaule of row was found

```
EQ0:  A    #ZeqFound
      JC    EQ1

      L     #ColIndexLs
      L     1
      -I
      L     #NoOfRow
      *I
      L     #RowIndexEq
      +I
      L     #IndexOffset
      +I
      L     4
      *I
      SLD   3
      LAR1
      L     DBD [AR1,P#0.0]
      T     #yC1

      L     #ColIndexGr
      L     1
      -I
      L     #NoOfRow
      *I
      L     #RowIndexEq
      +I
      L     #IndexOffset
      +I
      L     4
      *I
      SLD   3
      LAR1
      L     DBD [AR1,P#0.0]
      T     #yC2

      JU    N5E
```



```

EQ1:  A      #XeqFound
      A      #ZeqFound
      JC     EQ2

      L      #ColIndexEq
      L      1
      -I
      L      #NoOfRow
      *I
      L      #RowIndexLs
      +I
      L      #IndexOffset
      +I
      L      4
      *I
      SLD    3
      LAR1
      L      DBD [AR1,P#0.0]
      T      #yR1

      L      #ColIndexEq
      L      1
      -I
      L      #NoOfRow
      *I
      L      #RowIndexGr
      +I
      L      #IndexOffset
      +I
      L      4
      *I
      SLD    3
      LAR1
      L      DBD [AR1,P#0.0]
      T      #yR2

      JU     N5E

```

```

EQ2:  L    #ColIndexEq
      L    1
      -I
      L    #NoOfRow
      *I
      L    #RowIndexEq
      +I
      L    #IndexOffset
      +I
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #Result
      JU    N6E

```

```

N5E:  NOP  0

```

Network 6: Calculation

Output calculation

```

A    #XeqFound
O    #ZeqFound
JC   RES0

```

```
//For Y in ZLs
//Yzls = (((Inval - XLs) * deltaY) / deltaX) + y11
```

```

L      #y21
L      #y11
-R
T      #DeltaY

L      #XGr
L      #XLs
-R
T      #DeltaX

L      #XVal
L      #XLs
-R
L      #DeltaY
*R
L      #DeltaX
/R
L      #y11
+R
T      #YLs

```

```
//For Y in ZGr
//YzGr = (((Inval - XLs) * deltaY) / deltaX) + y12
```

```

L      #y22
L      #y12
-R
T      #DeltaY

L      #XGr
L      #XLs
-R
T      #DeltaX

L      #XVal
L      #XLs
-R
L      #DeltaY
*R
L      #DeltaX
/R
L      #y12
+R
T      #YGr

```

```
//For Y
//Y = (((Inval - XLs) * deltaY) / deltaX) + YLs
```

```
L    #YGr
L    #YLs
-R
T    #DeltaY
```

```
L    #ZGr
L    #ZLs
-R
T    #DeltaX
```

```
L    #ZVal
L    #ZLs
-R
L    #DeltaY
*R
L    #DeltaX
/R
L    #YLs
+R
T    #Result
```

```
JU    N6E
```

```
//If equal value of row was found
//Y = (((Mval - XLs) * deltaY) / deltaX) + yC1
```

```
RES0: A    #ZeqFound
      JC    RES1
```

```
L    #yC2
L    #yC1
-R
T    #DeltaY
```

```
L    #ZGr
L    #ZLs
-R
T    #DeltaX
```

```
L    #ZVal
L    #ZLs
-R
L    #DeltaY
*R
L    #DeltaX
/R
L    #yC1
+R
T    #Result
```

```
JU    N6E
```

```
//if equal value of column was found
//Y = (((Inval - XLs) * deltaY) / deltaX) + yR1
```

```
RES1: A    #XeqFound
      A    #ZeqFound
      JC   RES3
```

```
      L    #yR2
      L    #yR1
      -R
      T    #DeltaY
```

```
      L    #XGr
      L    #XLs
      -R
      T    #DeltaX
```

```
      L    #XVal
      L    #XLs
      -R
      L    #DeltaY
      *R
      L    #DeltaX
      /R
      L    #yR1
      +R
      T    #Result
```

```
JU    N6E
```

## Network 7: Writing output

Writing result of calculation to the output

A    #XeqFound  
O    #ZeqFound  
JC   FIN0

L    #XLs  
T    #XLsValue

L    #XGr  
T    #XGrValue

L    #ZLs  
T    #ZLsValue

L    #ZGr  
T    #ZGrValue

L    #y11  
T    #YLsLs

L    #y12  
T    #YLsGr

L    #y21  
T    #YGrLs

L    #y22  
T    #YGrGr

JU   N7E

FIN0:	A	#ZeqFound
	JC	FIN1
	L	#XEq
	T	#XLsValue
	L	#XEq
	T	#XGrValue
	L	#ZLs
	T	#ZLsValue
	L	#ZGr
	T	#ZGrValue
	L	#yC1
	T	#YLsLs
	L	#yC2
	T	#YLsGr
	L	#yC1
	T	#YGrLs
	L	#yC2
	T	#YGrGr
	JU	N7E

FIN1:	A	#XeqFound
	A	#ZeqFound
	JC	FIN2
	L	#XLs
	T	#XLsValue
	L	#XGr
	T	#XGrValue
	L	#ZEq
	T	#ZLsValue
	L	#ZEq
	T	#ZGrValue
	L	#yR1
	T	#YLsLs
	L	#yR2
	T	#YLsGr
	L	#yR1
	T	#YGrLs
	L	#yR2
	T	#YGrGr
	JU	N7E



```

FIN2: L    #RowIndexEq
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #XLsValue
      T    #XGrValue

      L    #ColIndexEq
      L    #NoOfRow
      +I
      L    4
      *I
      SLD  3
      LAR1
      L    DBD [AR1,P#0.0]
      T    #ZLsValue
      T    #ZGrValue

      L    #Result
      T    #YLsLs

      L    #Result
      T    #YLsGr

      L    #Result
      T    #YGrLs

      L    #Result
      T    #YGrGr

N7E:  L    #Result
      T    #Out

```

## Příloha II: Struktury použité v DB s daty pro linearizační funkci

### a) Struktura XValues

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	X1	REAL	0.000000e+000
+4.0	X2	REAL	5.000000e-001
+8.0	X3	REAL	1.000000e+000
+12.0	X4	REAL	1.500000e+000
+16.0	X5	REAL	2.000000e+000
+20.0	X6	REAL	2.500000e+000
+24.0	X7	REAL	3.000000e+000
+28.0	X8	REAL	3.500000e+000
=32.0		END_STRUCT	

### b) Struktura ZValues

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	Z1	REAL	0.000000e+000
+4.0	Z2	REAL	2.500000e+001
+8.0	Z3	REAL	5.000000e+001
+12.0	Z4	REAL	7.500000e+001
+16.0	Z5	REAL	1.000000e+002
+20.0	Z6	REAL	1.500000e+002
=24.0		END_STRUCT	

c) Struktura YValues

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	Y0_0	REAL	1.000000e+002
+4.0	Y0_05	REAL	1.000000e+002
+8.0	Y0_1	REAL	1.000000e+002
+12.0	Y0_15	REAL	1.000000e+002
+16.0	Y0_2	REAL	1.000000e+002
+20.0	Y0_25	REAL	1.000000e+002
+24.0	Y0_3	REAL	1.000000e+002
+28.0	Y0_35	REAL	1.000000e+002
+32.0	Y25_0	REAL	0.000000e+000
+36.0	Y25_05	REAL	8.230000e+000
+40.0	Y25_1	REAL	1.923000e+001
+44.0	Y25_15	REAL	3.736000e+001
+48.0	Y25_2	REAL	1.000000e+002
+52.0	Y25_25	REAL	1.000000e+002
+56.0	Y25_3	REAL	1.000000e+002
+60.0	Y25_35	REAL	1.000000e+002
+64.0	Y50_0	REAL	0.000000e+000
+68.0	Y50_05	REAL	5.960000e+000
+72.0	Y50_1	REAL	1.264000e+001
+76.0	Y50_15	REAL	2.036000e+001
+80.0	Y50_2	REAL	3.253000e+001
+84.0	Y50_25	REAL	6.541000e+001
+88.0	Y50_3	REAL	1.000000e+002
+92.0	Y50_35	REAL	1.000000e+002
+96.0	Y75_0	REAL	0.000000e+000
+100.0	Y75_05	REAL	4.790000e+000
+104.0	Y75_1	REAL	9.590000e+000
+108.0	Y75_15	REAL	1.574000e+001
+112.0	Y75_2	REAL	2.209000e+001
+116.0	Y75_25	REAL	3.258000e+001
+120.0	Y75_3	REAL	4.712000e+001
+124.0	Y75_35	REAL	1.000000e+002

+128.0	Y100_0	REAL	0.000000e+000
+132.0	Y100_05	REAL	4.090000e+000
+136.0	Y100_1	REAL	8.190000e+000
+140.0	Y100_15	REAL	1.295000e+001
+144.0	Y100_2	REAL	1.816000e+001
+148.0	Y100_25	REAL	2.404000e+001
+152.0	Y100_3	REAL	3.405000e+001
+156.0	Y100_35	REAL	4.655000e+001
+160.0	Y150_0	REAL	0.000000e+000
+164.0	Y150_05	REAL	3.340000e+000
+168.0	Y150_1	REAL	6.690000e+000
+172.0	Y150_15	REAL	1.005000e+001
+176.0	Y150_2	REAL	1.416000e+001
+180.0	Y150_25	REAL	1.905000e+001
+184.0	Y150_3	REAL	2.377000e+001
+188.0	Y150_35	REAL	3.225000e+001
=192.0		END_STRUCT	

### **Příloha III: Obsah přiloženého CD**

- Text práce ve formátu DOCX
- Text práce ve formátu PDF
- Úprava naměřených dat
- Fotografie